Videojuego para aprender a estimar raciones de carbohidratos

Trabajo de Fin de Grado

Grado en Ingeniería de Software

Laura Jiménez Fernández
Mario Rodríguez Salinero

Dirigido por
José Ignacio Hidalgo Pérez

Universidad Complutense de Madrid

Septiembre 2019

Documento maquetado con VIKTEX 1.0 basado en TEXIS

Este documento está preparado para ser impreso a doble cara.

Resumen

Every day it gets a little easier... But you gotta do it every day, that's the hard part.

Bojack Horseman

La insulina es la hormona producida por el páncreas para ayudar a que la glucosa que ingerimos a través de los alimentos se procese en el organismo y sea transportada a nuestras células para así suministrarles energía. La diabetes es una enfermedad por la cual el organismo no es capaz de procesar eficazmente o generar la insulina suficiente para completar este proceso. Esto provoca que, en ocasiones, los niveles de glucosa en la sangre sean inadecuados, puesto que ésta no es distribuida de una manera adecuada a través del torrente sanguíneo. La glucosa elevada puede ser perjudicial especialmente para el corazón, los riñones, y las arterias. Por ello, es esencial conocer las cantidades de glucosa que ingiere un paciente con diabetes, y la cantidad de insulina necesaria para mantener unos niveles adecuados de glucosa en sangre, en aquellos pacientes que deban suministrarse insulina exógena. Ahora bien: ¿cómo podemos saber cuál es la cantidad de glucosa que ingerimos? Antes de responder a esta pregunta, es importante saber que la glucosa proviene de los hidratos de carbono que consumimos y su absorción. Habitualmente, las personas que sufren diabetes estiman la cantidad de hidratos de carbono que consumen mediante una medida denominada ración de carbohidratos que equivale a 10 g. En este trabajo de fin de grado hemos desarrollado un videojuego, a su vez dividido en dos juegos diferentes, que facilitará a las personas con diabetes el aprendizaje del cálculo de las raciones de carbohidratos que tienen los alimentos que van a digerir en su día a día. Además, podrán mejorar la comprensión del cálculo de las dosis de insulina, y observar cómo éstas repercuten en la glucosa.

Palabras Clave: Diabetes, Glucosa, Carbohidratos, Insulina, Videojuego, Unity, C#.

Abstract

I invent, transform, create and destroy for a living and when I don't like something about the World I change it.

Rick and Morty

Insulin is an hormone produced by the pancreas to help the glucose we eat through food to be processed in the body and transported to our cells in order to provide them energy. Diabetes is a desease whereby the organism is not capable of process efficiently or generate enough insuline to complete this process. This causes that sometimes blood glucose levels are inadequate, since it is not distributed properly through the bloodstream. High blood glucose can be harmful, especially to heart, kidneys, and arteries. That is why it is essential to know the amount of glucose ingested by a patient of diabetes and the amount of insulin needed to mainaint adequate blood glucose levels, in those patiens who must be given exogeous insulin. But how can we know the amount of glucose we eat? It would therefore be important to know that glucose comes from the carbohydrates that we eat and its absorption. Normally, people with diabetes estimate the amount of carbohydrates that they eat through a measure named carbohydrates ration which is equivalent to 10g. In this final degree project we have developed a videogame that is further divided into two different games, which will certainly make it easier to the patient with diabetes the learn of calculation of carbohydrate rations that have the food they will digest in their everyday life. In addition, they will be able to improve the understanding of the calculation of insulin doses, and observe how they affect glucose.

Keywords: Diabetes, Glucose, Carbohydrates, Insulin, Videogame, Unity, C#.

Índice general

Capítulo 1

Introducción

Mucha gente pequeña, en lugares pequeños, haciendo cosas pequeñas, puede cambiar el mundo.

Eduardo Galeano

Actualmente la existencia y el avance tan veloz de las nuevas tecnologías están permitiendo que nuestro estilo de vida haya mejorado, facilitando muchas de las tareas de la vida cotidiana. Se desarrollan constantemente nuevas plataformas y aplicaciones sin las que ya nos resulta impensable vivir. Este trabajo surgió con la idea de proporcionar una herramienta para ayudar a los pacientes diabéticos en la autogestión de su enfermedad, favoreciendo el conocimiento de las raciones de carbohidratos en la alimentación en su relación con la glucosa y la insulina. De esta manera, se complementa con los medios convencionales de información sobre la alimentación y el ejercicio físico en la diabetes de una manera más entretenida y a su vez forma al respecto. Existen varios tipos de diabetes, y aunque seguramente este proyecto será de mayor utilidad a las personas con diabetes tipo 1, el videojuego es útil para cualquier persona que comience con esta enfermedad. Nuestra motivación va muy ligada a la idea de querer mejorar la vida de las personas a través del desarrollo de software.

Debido a nuestro interés en el mundo del desarrollo de los videojuegos, y al potencial que vimos en este proyecto, decidimos apostar por éste. Gracias al esfuerzo realizado a lo largo de este año, y a todas las personas que han contribuido en el proyecto, ha sido posible finalizar la primera implementación de un videojuego que puede ser de utilidad para centros sanitarios, familias y pacientes con diabetes.

La diabetes tiene un elevado impacto en la población: según el informe mundial sobre la diabetes de la OMS, de abril de 2016 (1), se estimó que 422 millones de personas en todo el mundo tenían diabetes en 2014, frente a los 108 millones en 1980. La prevalencia mundial de la diabetes casi se ha duplicado durante dicho periodo de tiempo, pasando del 4,7 % al 8,5 % en las personas adultas. En 2012, la diabetes provocó 1,5 millones de muertes, así como un nivel de glucosa en sangre superior al

recomendado provocó otros 2,2 millones de fallecimientos,al incrementar el riesgo de problemas cardiovasculares.

Los videojuegos pueden servir como dispositivos de socialización y educación, aunque no estén ligados a la educación formal (2), de hecho, existe una categoría denominada "juegos serios", que son aquellos que se centran en educar, entrenar y enseñar, dejando en un segundo plano el aspecto lúdico que éstos tienen habitualmente (3). En el caso concreto de nuestro videojuego serio, se pretende llegar de una manera lúdica al mayor número posible de personas con diabetes.

1.1. Objetivos

Nuestro objetivo principal en este trabajo de fin de grado ha sido desarrollar un videojuego, para ayudar a pacientes con diabetes a estimar las raciones de carbohidratos que tienen los alimentos que ingieren. El juego está dividido en dos partes, y ha sido diseñado y desarrollado completamente en el ámbito de este TFG. Para cumplir dicho objetivo, se han planteado los siguientes objetivos específicos:

- Estudio bibliográfico de la diabetes.
- Aprendizaje sobre Unity y C#.
- Elaboración de sprites (pixelArt) que servirán para los juegos, así como la obtención de sprites gratuitos.
- Creación de un prototipo del juego de plataformas y otro del juego de preguntas utilizando los sprites creados.
- Unificación de los prototipos a través de un menú principal.
- Creación del juego de plataformas y juego de preguntas definitivos en base a los prototipos.
- Demostrar la utilidad del videojuego (objetivo general).

1.2. Plan de trabajo

Para establecer un plan de trabajo adecuado, decidimos basarnos en la metodología ágil SCRUM. La planificación seguida en este trabajo comenzó con la búsqueda de información y documentación en diferentes aspectos. Inicialmente nos informamos sobre la diabetes, y en concreto sobre la diabetes de tipo 1: qué es, cómo afecta, y cómo se trata.

Después, hicimos una captura de requisitos inicial con nuestro director del trabajo de fin de grado. Definimos cómo deberían quedar los juegos a través de Historias de Usuario, y con la técnica MOSCOW, para así establecer prioridades a la hora de desarrollar los juegos: cuáles serían los requisitos para hacer el juego funcional, qué

implementaciones serían necesarias, posibles mejoras para el juego en un futuro.
Por una parte, se planteó un juego de plataformas en 2D en el que el personaje que maneja el jugador irá avanzando por un nivel cogiendo alimentos e insulina y consumiéndola cuando sea necesario, para así poder controlar el nivel de glucosa en sangre y evitar complicaciones como el shock hipoglucemico e hiperglucémico. Por otra parte, se ha desarrollado un juego de preguntas dividido en niveles en el que se muestran imágenes de raciones de alimentos, y para ir avanzando por los niveles, el jugador tendrá que acertar cuántas raciones de carbohidratos equivale a cada ración de comida.

Tras la elaboración de las Historias de Usuario, probamos diversas herramientas para decidir qué usaríamos como motor gráfico del juego, en qué lenguaje programaríamos, y qué usaríamos para crear los Sprites necesarios. Finalmente, decidimos usar Unity, Visual Studio para desarrollar en C#, y Gimp para crear y editar los sprites. Posteriormente iniciamos nuestra formación en Unity y C# a través de cursos online, libros, documentos técnicos y tutorías con profesores del Grado de Videojuegos de la Universidad Complutense de Madrid para resolución de dudas.

Paralelamente diseñamos algunos mockups y diagramas de clase para aclarar nuestra visión sobre los juegos, ayudándonos a desarrollar varios prototipos de ambos juegos por separado para así aplicar nuestros conocimientos aprendidos con Unity y C#.

Finalmente, diseñamos un prototipo definitivo de cada juego que revisamos con nuestro director del trabajo de fin de grado. Los prototipos nos sirvieron para crear un diseño básico con diagramas de clases para posteriormente saber cómo estructurar y desarrollar el videojuego definitivo.

Después procedimos a unificar los prototipos, y aumentar las funcionalidades de cada juego, hasta terminar obteniendo el videojuego definitivo. A medida que iba avanzando el desarrollo, pedimos a personas de nuestro entorno que probaran el videojuego para así comprobar la jugabilidad, curva de dificultad y detección de áreas de mejora.

1.3. Organización de la memoria

El resto de esta memoria está organizada de la siguiente manera:

- En el capítulo 2 hacemos una breve introducción a la diabetes: definición, tipos, cómo afecta al paciente, y su importancia en nuestro proyecto.
- En el capítulo 3 entramos en detalles más técnicos: el diseño del software. Nos centramos en la captura de requisitos, selección de herramientas y en la elaboración de mockups y diagramas.
- En el capítulo 4 explicamos cómo ha sido todo el proceso del desarrollo del software: desde la creación de prototipos, hasta llegar a la versión final.

- En el capítulo 5 se podrán encontrar los resultados finales del proyecto.
- Por último, en el capítulo 6 se incluyen las conclusiones de nuestro proyecto y posibles mejoras que planteamos sobre este videojuego.

Introduction

There's a lot of beauty in ordinary things.
Isn't that kind of the point?

The Office

Nowadays, the existence and the rapid progress of new technologies allows our lyfestyle to improve, facilitating many of our daily life tasks. It is already unthinkable for us to live without any new platforms and applications that are constantly being developed. This project arose from the idea of providing a tool to help diabetic patients in the self-management of their illness, thereby promoting the knowledge of carbohydrates rations in their daily diet and its relation with glucose and insulin. This complements the conventional means of information about diet and physical exercise in diabetes in a more entertaining way, and, in turn, it teaches about it. There are several types of diabetes, and though this project will be most useful for people who have type one diabetes, the videogame is helpful for anyone starts with this disease. Our motivation is closely linked with the idea of improving people's lives trhough software development.

Because of our interest in the world of videogame development and the potential we saw in this project, we decided to bet on this one. Thanks to the efforts invested throughout the year and to the people that contributed in this project, it has been possible to finalise the first implementation of a videogame that can be useful to health centres, families and patients with diabetes.

Diabetes has a heavy global impact: OMS' World Report on Diabetes, issued in April, 2016, (1) estimated that 422 million people all over the world had diabetes in 2014, compared with 108 million people that had diabetes in 1980. The global prevalence of diabetes has been almost doubled during that period of time: it has increased from 4.7 % to 8.5 % in adults. In addition, there was 1.5 millions of deaths caused by diabetes in 2012. A blood glucose level higher than recommended caused other 2,2 million deaths, since it increases the risk of cardiovascular illnesses.

Videogames can be used as socialization and educational tools, even though they are not related with the formal education. In fact, there is a videogames category named 'serious games', whose are those focused on education, training and learning, leaving in a second state the playful aspect that videogames have usually (3). Th-rough this serious game, it's pretended to reach the largest number of patients with

diabetes in a playful way.

1.4. Objetives

Our principal objective in this final degree project have been developing a videogame, to help patients with diabetes to estimate the carbohydrates rations that the food that they are eating has. The videogame is divided in two parts, and it was designed and developed totally in this final degree project.

For reaching this principal objective, we raised the following specific objectives:

- Bibliographic study of the diabetes.
- Unity and C# learning.
- Sprites elaboration (pixelArt) whose will be useful for the games, and the obtainment of free open sprites.
- Creation of the platform prototype and quiz prototype using the generated sprites.
- Unification of the prototypes using a main menu.
- Creation of the platform and quiz final games based on the prototypes.
- Demonstrate the videogame utility (general objective).

1.5. Work plan

For establishing a proper work plan, we decided to focus on the SCRUM agile methodology. The planning followed during this project started with the research of the information and documentation in different aspects. Initially, we focused on diabetes, specifically in diabetes type 1: how this is, how it affects to the pacients, and how it is treated.

We defined the original requirements of the project with our mentor. We had to decide how the games should be through the user stories applying the MOSCOW method in order to establish the priorities for developing them, what were the conditions to create the functional set, the implementations that were necessary and future improvement.

The methodology used to develop the games was based on:

- Relying on 2D platform game, there is a character controlled by the player. While the character is moving through the level he needs to control his glycemia levels. In order to do that he needs to consume food and insulin if necessary to avoid hypoglycemia or hyperglycemia and their health implications.

- We also made a more pedagogic question and answer quiz game that is divided according to images about food rations. The player has to guess the grams of carbohydrates equivalent in each food ration.

After the elaboration of the User Stories, we settled on the tools and technologies to use, such as the graphic engine, programming language and the required artwork. We agreed on using Unity, Visual Studio to develop in C# and Gimp to create and edit the Sprites.

Afterwards we started the formation in Unity and C# thanks to online courses, books, technique documents, some mentorships with teachers from the videogames bachelor's degree to solve some of our doubts.

At the same time we designed some mockups and diagrams to clear our perspectives about the videogame. We could built some isolated prototypes up, applying the knowledge that we learned in Unity and C#.

Finally, we designed the final prototype for each game revising it with our mentor. Thanks to these prototypes, we created a basic design using UML's Class Diagrams to know how to carry on with the final videogame.

We finished the project by consolidating the prototypes and increasing the functionality of each game. Before the final release we asked people around us to beta test the game for us. Their findings and feedback helped us gather valuable information about the gameplay, the difficulty and what needed improving.

1.6. Report organization

The report is organized as follows:

- Chapter 2 provides a short introduction to diabetes: definition, types, how it affects the patient, and its importance in our project.
- Chapter 3 reviews some technical details of software design: gathering of requirements, selection of tools, elaboration of mockups and diagrams.
- Chapter 4 provides an explanation of the software development process: from prototyping to the final version.
- Chapter 5 provides the final results of the project.
- Finally, conclusions and possible improvements are presented in chapter 6.

Capítulo 2

Diabetes

You can live with diabetes. It's not the worst thing to have, but you have to manage yourself and have some self control.

Tony Rock

En este capítulo explicamos a fondo un concepto que hemos introducido previamente: la diabetes. Se podrá encontrar la definición de ésta, y además, introduciremos los tipos de diabetes que existen, aunque únicamente nos vayamos a centrar en un tipo durante nuestro videojuego.

2.1. La diabetes

La diabetes es una enfermedad crónica que aparece cuando el páncreas o bien no produce insulina suficiente o cuando el organismo no utiliza de manera eficaz la que éste produce (4). Nuestro organismo está formado por millones de células que necesitan proveerse de energía. Dicha energía la obtienen de la glucosa, la cual se obtiene a través de alimentos que contengan hidratos de carbono. Para que ésta pueda entrar en las células es necesario que las propias células dispongan de receptores suficientes, y que haya suficiente cantidad de insulina para que ésta se acople a los receptores y haga posible que la glucosa pueda entrar en las células (5).

La insulina es una hormona producida en el páncreas, y, como hemos mencionado, ayuda a que la glucosa sea transportada a las células y les suministre a éstas energía. Cuando las células tienen receptores e insulina suficientes, las células producen energía con normalidad, metabolizando la glucosa. Sin embargo, sin suficiente insulina, la glucosa permanece en la sangre. Por ello, si la diabetes no se controla, el paciente puede padecer hiperglucemia, es decir tener niveles altos de glucosa en la sangre, lo cual, de ser algo habitual, puede provocar alteraciones en la función de diversos órganos, especialmente los ojos, los riñones, los nervios, el corazón y los vasos sanguíneos (6).

La diabetes es una enfermedad muy común: en 2014 la OMS estimaba que había 422 millones de personas con diabetes en el mundo (1). Además, ésta considera que el 50 % de las personas que tienen diabetes están sin diagnosticar.

2.2. Tipos de diabetes

2.2.1. Diabetes tipo 1

La diabetes tipo 1 tipo debuta principalmente en niños, aunque también puede darse en adolescentes y adultos. Suele aparecer de forma brusca, normalmente independiente de antecedentes familiares, y no está ligada con el estilo de vida de la persona en la que debuta.

Su particularidad es que las células del páncreas que producen la insulina son destruidas por autoanticuerpos, es decir, se origina por un proceso autoinmune (el organismo ataca a las propias células como si fueran extrañas) (7). Los pacientes que presentan diabetes de tipo 1 deben tratarse con varias dosis diarias de insulina, ya sea mediante múltiples dosis inyectadas manualmente, o con la ayuda de una bomba de insulina. De cualquier modo, deberán calcular la dosis diaria de insulina que necesitan.

Independientemente del tipo de diabetes que tiene cada paciente en particular, hacer actividad física con regularidad es importante para su salud y bienestar general. No obstante, con la diabetes tipo 1, es muy importante usar la dosis de insulina que corresponde según los alimentos que ingiere a lo largo del día y la actividad física que realiza, puesto que se puede prevenir el desarrollo de complicaciones en este tipo de diabetes.

Además, tomó fuerza la importancia de administrar las dosis de insulina de una manera lo más fisiológica y flexible posible, lo que nos lleva a hablar de la terapia bolo-basal. En esta se le administra al paciente una insulina de acción prolongada que cubre las necesidades de insulina entre ingestas de comida, junto con la administración de otro tipo de insulina, esta vez de acción rápida y corta duración, la cual se administra en las propias comidas, bien para cubrir los carbohidratos de las mismas, o para corregir una situación de hiperglucemia. Dentro de dicha terapia la práctica más común en nuestro país es la administración de múltiples dosis de insulina tanto de acción prolongada, como de acción rápida (8).

La dosificación de insulina prolongada suele ser fija, puesto que las necesidades basales de insulina de los pacientes apenas varían de un día a otro. Sin embargo, sí que se presentan ciertas complicaciones a la hora de calcular la dosis de insulina prandial, es decir, la insulina de acción rápida, puesto que requieren un cálculo en cada momento en función de los carbohidratos de la comida que se ha ingerido, de la actividad física que se va a realizar, el valor de glucemia capilar, y las necesidades de insulina del individuo en función del momento del día. Por ello, se necesitan conocer datos como la ración de carbohidratos de una ingesta de comida, conocer el ratio de insulina/carbohidratos (unidades de insulina que cubre una ración de hidratos de

carbono), en cada ingesta y aplicar el factor de sensibilidad a la insulina (reducción esperable de glucemia capilar en mg/dl por 1 unidad de insulina) para corregir un valor de glucemia por encima del objetivo.

2.2.2. Diabetes tipo 2

La forma más frecuente de diabetes es la diabetes tipo 2 en el 85-90 % de los casos (9). Es un tipo de diabetes casi exclusiva de adultos, aunque actualmente se va viendo cada vez más en niños y adolescentes. En EEUU, de cada 10 diabéticos menores de 18 años, 2 ó 3 tienen diabetes tipo 2. Ésta se caracteriza porque el cuerpo del paciente no utiliza correctamente la propia insulina. Las personas adultas, con sobrepeso, historial familiar de diabetes, y con un estilo de vida sedentario, tienen un riesgo mayor de tener diabetes. Sufrir prediabetes (10) también se considera un factor de riesgo.

Se considera que tienen prediabetes aquellas personas que tienen niveles de glucosa en la sangre más alto de lo normal, pero no lo suficiente para que sean consideradas diabéticas. Si una persona está en riesgo de padecer diabetes de tipo 2, se puede retrasar o prevenir su desarrollo haciendo unos cambios en el estilo de vida, como una dieta más saludable y haciendo ejercicio frecuentemente.

Los síntomas de la diabetes tipo 2 pueden ser similares a los de la diabetes de tipo 1, pero a menudo menos intensos. Por ello, ésta puede diagnosticarse en los pacientes cuando ya tiene varios años de evolución y aparecen complicaciones de la misma.

2.3. Cómo afecta la diabetes al paciente

Para evitar que la glucosa le suba o baje demasiado al individuo con diabetes, es importante planificar con tiempo y saber cómo le suele responder la glucosa en la sangre al ejercicio, así como las raciones de carbohidratos que ingiere en cada comida. Además, el paciente puede notar síntomas tales como:

- Aumento de sed y de ganas de orinar
- Fatiga
- Aumento del apetito
- Pérdida de peso sin razón aparente
- Visión borrosa
- Entumecimiento u hormigueo en manos o pies
- Úlceras que no cicatrizan

Dichos síntomas pueden afectar en gran medida al estilo de vida de la persona en cuestión.

2.4. Importancia de la diabetes en nuestro videojuego

Como hemos explicado en 2.2.1, las personas con diabetes necesitan hacer una estimación de las raciones de carbohidratos que van a ingerir en las comidas, conocer el ratio de insulina/carbohidratos, y aplicar el factor de sensibilidad a la insulina, así como la cantidad de ejercicio que van a realizar, aunque sea en tareas cotidianas. Este cálculo tiene cierta complejidad, puesto que influyen diversas variables a tener en cuenta.

Nuestro videojuego se ha realizado con el fin de facilitar el aprendizaje de dichos cálculos. No obstante, empezamos con una implementación lineal de dichos cálculos para comprobar que funciona. En el juego de plataformas, se tiene en cuenta que cada alimento sube una cantidad determinada de glucosa, y con una velocidad de absorción determinada. En cuanto a la insulina prandial, sí que pusimos que baja una cantidad fija de glucosa (pues se administra siempre la misma cantidad, de una en una), y lo mismo hicimos con el tiempo de absorción. Además, cuando el personaje se mueve por el nivel, a éste le va bajando la glucosa progresivamente. Esta última implementación tiene como objetivo simular el ejercicio que los pacientes con diabetes realizan en su día a día.

El juego de preguntas es mucho más simple en cuanto a los cálculos se refiere, puesto que no hay que hacer tantas estimaciones. Solo hay que calcular cuántas raciones de carbohidratos (según la medida estándar vigente en España (11), por la cual 1 ración de carbohidratos es igual a 10 gramos de hidratos de carbono) tienen las comidas que se muestran en la imagen de cada pregunta. La respuesta correcta es una estimación que hemos hecho con la calculadora de carbohidratos del hospital de Barcelona, y como se comentará más adelante, en 4.1.2, el resto de respuestas son valores generados aleatoriamente que se acercan a la respuesta correcta.

Capítulo 3

Diseño del software

We all make choices,
but in the end, our choices make us.

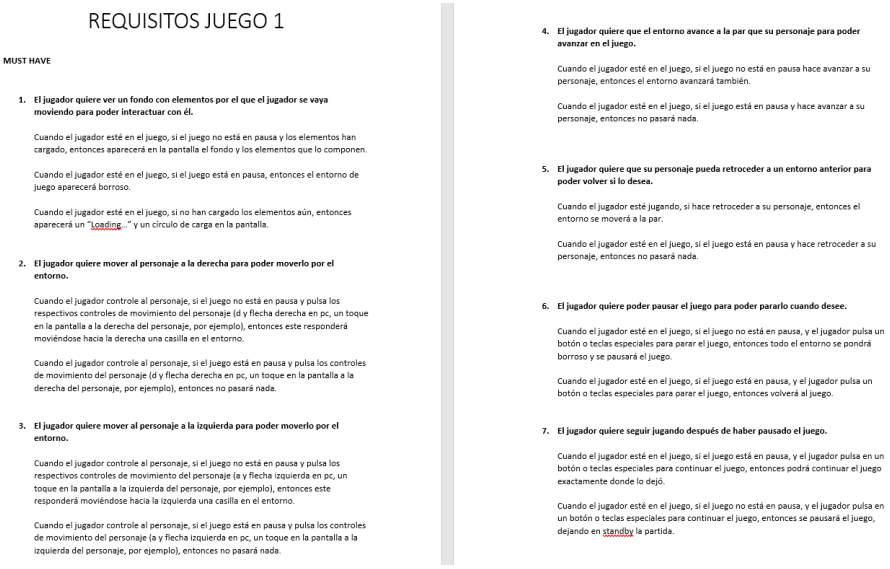
Bioshock

Uno de los grandes problemas que se aprecian a día de hoy durante el ciclo de vida del software es la mala definición o una definición ambigua de los requisitos del producto, así como un mal diseño de software que hace que este sea poco mantenible. Se ha extendido en empresas que utilizan tanto metodologías ágiles, como tradicionales y puede ser - y es - crítico para los proyectos. Para así evitar este problema, decidimos basar todo el proceso de desarrollo del videojuego en una metodología a la que pudiéramos adaptarnos por nuestro tipo de proyecto. Por ello, decidimos utilizar SCRUM. Las metodologías ágiles son muy usadas en el mundo del desarrollo de los videojuegos, y esta en concreto es una con la que ya nos sentíamos familiarizados, puesto que ya la habíamos realizado proyectos con dicha metodología previamente.

3.1. Captura de requisitos

La primera fase consistió en reunirnos con nuestro director de trabajo de fin de grado para así poder hacer una captura completa de requisitos. En dicha reunión desarrollamos las Historias de Usuario de cada juego con detalle, ordenándolas según la prioridad de cada una, basándonos en la técnica de priorización MoSCoW.

En MoSCoW se dividen las Historias de Usuario en *Must Have*, *Should Have*, *Could Have*, y *Would like but won't get*. La clasificación más prioritaria, *Must Have*, comprende los requisitos que tienen que estar implementados para que el producto tenga la funcionalidad mínima. El *Should Have* abarca los requisitos importantes del producto, pero que no son obligatorios como en el caso del Must Have. En cuanto al *Could Have*, comprende Historias de Usuario que podrían implementarse. Y por último, tenemos la clasificación menos prioritaria, el *Would like but won't get*, la cual abarca requisitos que de momento no se implementarán, pero que podrían hacerse



REQUISITOS JUEGO 1

MUST HAVE

1. **El jugador quiere ver un fondo con elementos por el que el jugador se vaya moviendo para poder interactuar con él.**

   Cuando el jugador esté en el juego, si el juego no está en pausa y los elementos han cargado, entonces aparecerá en la pantalla el fondo y los elementos que lo componen.

   Cuando el jugador esté en el juego, si el juego está en pausa, entonces el entorno de juego aparecerá borroso.

   Cuando el jugador esté en el juego, si no han cargado los elementos aún, entonces aparecerá un "Loading.." y un círculo de carga en la pantalla.

2. **El jugador quiere mover al personaje a la derecha para poder moverlo por el entorno.**

   Cuando el jugador controle al personaje, si el juego no está en pausa y pulsa los respectivos controles de movimiento del personaje (d y flecha derecha en pc, un toque en la pantalla a la derecha del personaje, por ejemplo), entonces este responderá moviéndose hacia la derecha una casilla en el entorno.

   Cuando el jugador controle al personaje, si el juego está en pausa y pulsa los controles de movimiento del personaje (d y flecha derecha en pc, un toque en la pantalla a la derecha del personaje, por ejemplo), entonces no pasará nada.

3. **El jugador quiere mover al personaje a la izquierda para poder moverlo por el entorno.**

   Cuando el jugador controle al personaje, si el juego no está en pausa y pulsa los respectivos controles de movimiento del personaje (a y flecha izquierda en pc, un toque en la pantalla a la izquierda del personaje, por ejemplo), entonces este responderá moviéndose hacia la izquierda una casilla en el entorno.

   Cuando el jugador controle al personaje, si el juego está en pausa y pulsa los controles de movimiento del personaje (a y flecha izquierda en pc, un toque en la pantalla a la izquierda del personaje, por ejemplo), entonces no pasará nada.

4. **El jugador quiere que el entorno avance a la par que su personaje para poder avanzar en el juego.**

   Cuando el jugador esté en el juego, si el juego no está en pausa hace avanzar a su personaje, entonces el entorno avanzará también.

   Cuando el jugador esté en el juego, si el juego está en pausa y hace avanzar a su personaje, entonces no pasará nada.

5. **El jugador quiere que su personaje pueda retroceder a un entorno anterior para poder volver si lo desea.**

   Cuando el jugador esté jugando, si hace retroceder a su personaje, entonces el entorno se moverá a la par.

   Cuando el jugador esté en el juego, si el juego está en pausa y hace retroceder a su personaje, entonces no pasará nada.

6. **El jugador quiere poder pausar el juego para poder pararlo cuando desee.**

   Cuando el jugador esté en el juego, si el juego no está en pausa, y el jugador pulsa un botón o teclas especiales para parar el juego, entonces todo el entorno se pondrá borroso y se pausará el juego.

   Cuando el jugador esté en el juego, si el juego está en pausa, y el jugador pulsa un botón o teclas especiales para parar el juego, entonces volverá al juego.

7. **El jugador quiere seguir jugando después de haber pausado el juego.**

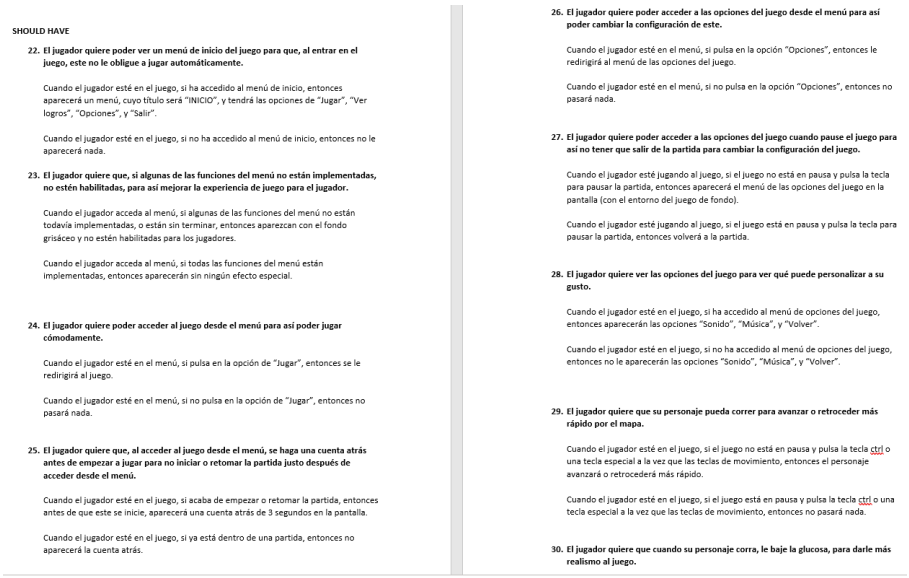
   Cuando el jugador esté en el juego, si el juego está en pausa, y el jugador pulsa en un botón o teclas especiales para continuar el juego, entonces podrá continuar el juego exactamente donde lo dejó.

   Cuando el jugador esté en el juego, si el juego no está en pausa, y el jugador pulsa en un botón o teclas especiales para continuar el juego, entonces se pausará el juego, dejando en standby la partida.

Figura 3.1: Ejemplo de requisitos del juego de plataformas que entran en la categoría *Must*

en un futuro.

Dividimos las Historias de Usuario en dos archivos diferentes, "Requisitos Juego 1" para el juego de plataformas, que se puede observar en las figuras 3.1, 3.2 y 3.3, y "Requisitos Juego 2" para el juego de preguntas, como también se puede ver en las figuras 3.4, 3.5, y 3.6. No incluimos ningún *Would like but won't get* de primeras, puesto que no había ninguna Historia de Usuario que consideráramos que tuviera tan baja prioridad.



SHOULD HAVE

22. **El jugador quiere poder ver un menú de inicio del juego para que, al entrar en el juego, este no le obligue a jugar automáticamente.**

    Cuando el jugador esté en el juego, si ha accedido al menú de inicio, entonces aparecerá un menú, cuyo título será "INICIO", y tendrá las opciones de "Jugar", "Ver logros", "Opciones", y "Salir".

    Cuando el jugador esté en el juego, si no ha accedido al menú de inicio, entonces no le aparecerá nada.

23. **El jugador quiere que, si algunas de las funciones del menú no están implementadas, no estén habilitadas, para así mejorar la experiencia de juego para el jugador.**

    Cuando el jugador acceda al menú, si algunas de las funciones del menú no están todavía implementadas, o están sin terminar, entonces aparezcan con el fondo grisáceo y no estén habilitadas para los jugadores.

    Cuando el jugador acceda al menú, si todas las funciones del menú están implementadas, entonces aparecerán sin ningún efecto especial.

24. **El jugador quiere poder acceder al juego desde el menú para así poder jugar cómodamente.**

    Cuando el jugador esté en el menú, si pulsa en la opción de "Jugar", entonces se le redirigirá al juego.

    Cuando el jugador esté en el menú, si no pulsa en la opción de "Jugar", entonces no pasará nada.

25. **El jugador quiere que, al acceder al juego desde el menú, se haga una cuenta atrás antes de empezar a jugar para no iniciar o retomar la partida justo después de acceder desde el menú.**

    Cuando el jugador esté en el juego, si acaba de empezar o retomar la partida, entonces antes de que este se inicie, aparecerá una cuenta atrás de 3 segundos en la pantalla.

    Cuando el jugador esté en el juego, si ya está dentro de una partida, entonces no aparecerá la cuenta atrás.

26. **El jugador quiere poder acceder a las opciones del juego desde el menú para así poder cambiar la configuración de este.**

    Cuando el jugador esté en el menú, si pulsa en la opción "Opciones", entonces le redirigirá al menú de las opciones del juego.

    Cuando el jugador esté en el menú, si no pulsa en la opción "Opciones", entonces no pasará nada.

27. **El jugador quiere poder acceder a las opciones del juego cuando pause el juego para así no tener que salir de la partida para cambiar la configuración del juego.**

    Cuando el jugador esté jugando al juego, si el juego no está en pausa y pulsa la tecla para pausar la partida, entonces aparecerá el menú de las opciones del juego en la pantalla (con el entorno del juego de fondo).

    Cuando el jugador esté jugando al juego, si el juego está en pausa y pulsa la tecla para pausar la partida, entonces volverá a la partida.

28. **El jugador quiere ver las opciones del juego para ver qué puede personalizar a su gusto.**

    Cuando el jugador esté en el juego, si ha accedido al menú de opciones del juego, entonces aparecerán las opciones "Sonido", "Música", y "Volver".

    Cuando el jugador esté en el juego, si no ha accedido al menú de opciones del juego, entonces no le aparecerán las opciones "Sonido", "Música", y "Volver".

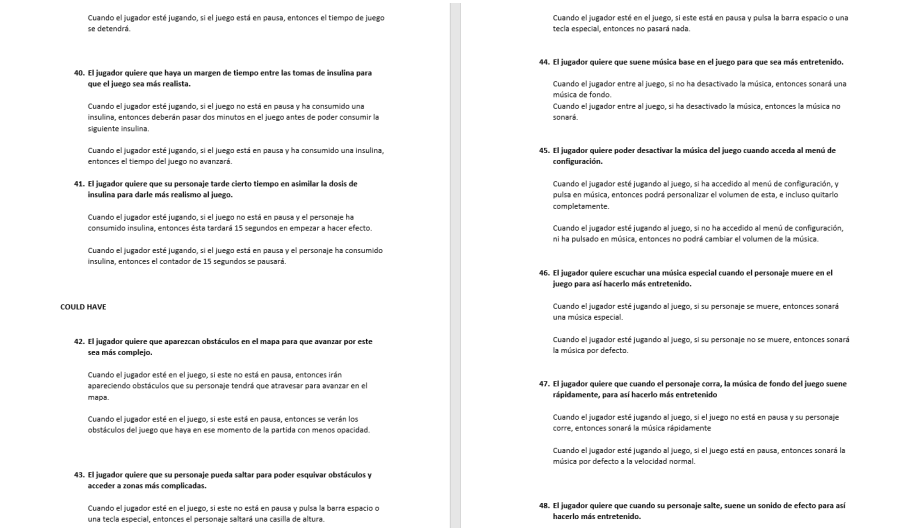
29. **El jugador quiere que su personaje pueda correr para avanzar o retroceder más rápido por el mapa.**

    Cuando el jugador esté en el juego, si el juego no está en pausa y pulsa la tecla ctrl o una tecla especial a la vez que las teclas de movimiento, entonces el personaje avanzará o retrocederá más rápido.

    Cuando el jugador esté en el juego, si el juego está en pausa y pulsa la tecla ctrl o una tecla especial a la vez que las teclas de movimiento, entonces no pasará nada.

30. **El jugador quiere que cuando su personaje corra, le baje la glucosa, para darle más realismo al juego.**

Figura 3.2: Ejemplo de requisitos del juego de plataformas que entran en la categoría *Should*



Cuando el jugador esté jugando, si el juego está en pausa, entonces el tiempo de juego se detendrá.

40. **El jugador quiere que haya un margen de tiempo entre las tomas de insulina para que el juego sea más realista.**

    Cuando el jugador esté jugando, si el juego no está en pausa y ha consumido una insulina, entonces deberán pasar dos minutos en el juego antes de poder consumir la siguiente insulina.

    Cuando el jugador esté jugando, si el juego está en pausa y ha consumido una insulina, entonces el tiempo del juego no avanzará.

41. **El jugador quiere que su personaje tarde cierto tiempo en asimilar la dosis de insulina para darle más realismo al juego.**

    Cuando el jugador esté jugando, si el juego no está en pausa y el personaje ha consumido insulina, entonces ésta tardará 15 segundos en empezar a hacer efecto.

    Cuando el jugador esté jugando, si el juego está en pausa y el personaje ha consumido insulina, entonces el contador de 15 segundos se pausará.

COULD HAVE

42. **El jugador quiere que aparezcan obstáculos en el mapa para que avanzar por este sea más complejo.**

    Cuando el jugador esté en el juego, si este no está en pausa, entonces irán apareciendo obstáculos que su personaje tendrá que atravesar para avanzar en el mapa.

    Cuando el jugador esté en el juego, si este está en pausa, entonces se verán los obstáculos del juego que haya en ese momento de la partida con menos opacidad.

43. **El jugador quiere que su personaje pueda saltar para poder esquivar obstáculos y acceder a zonas más complicadas.**

    Cuando el jugador esté en el juego, si este no está en pausa y pulsa la barra espacio o una tecla especial, entonces el personaje saltará una casilla de altura.

    Cuando el jugador esté en el juego, si este está en pausa y pulsa la barra espacio o una tecla especial, entonces no pasará nada.

44. **El jugador quiere que suene música base en el juego para que sea más entretenido.**

    Cuando el jugador entre al juego, si no ha desactivado la música, entonces sonará una música de fondo.
    Cuando el jugador entre al juego, si ha desactivado la música, entonces la música no sonará.

45. **El jugador quiere poder desactivar la música del juego cuando acceda al menú de configuración.**

    Cuando el jugador esté jugando al juego, si ha accedido al menú de configuración, y pulsa en música, entonces podrá personalizar el volumen de esta, e incluso quitarlo completamente.

    Cuando el jugador esté jugando al juego, si no ha accedido al menú de configuración, ni ha pulsado en música, entonces no podrá cambiar el volumen de la música.

46. **El jugador quiere escuchar una música especial cuando el personaje muere en el juego para así hacerlo más entretenido.**

    Cuando el jugador esté jugando al juego, si su personaje se muere, entonces sonará una música especial.

    Cuando el jugador esté jugando al juego, si su personaje no se muere, entonces sonará la música por defecto.

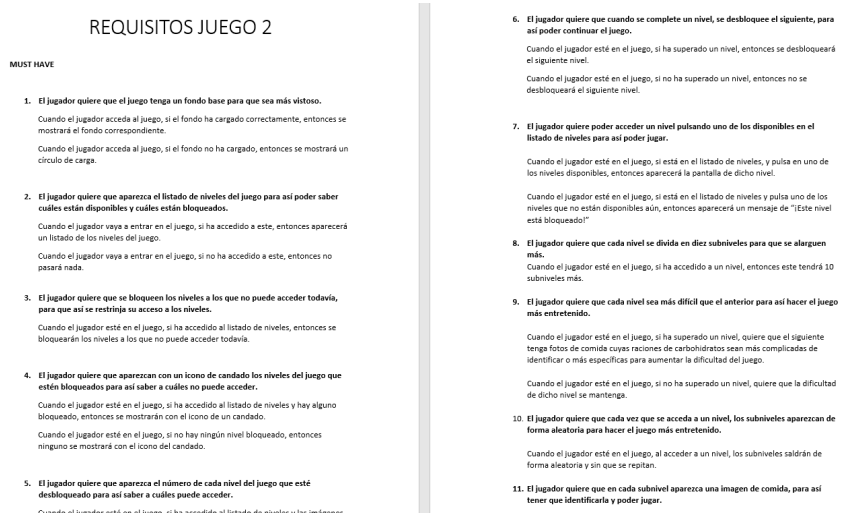
47. **El jugador quiere que cuando el personaje corra, la música de fondo del juego suene rápidamente, para así hacerlo más entretenido**

    Cuando el jugador esté jugando al juego, si el juego no está en pausa y su personaje corre, entonces sonará la música rápidamente

    Cuando el jugador esté jugando al juego, si el juego está en pausa, entonces sonará la música por defecto a la velocidad normal.

48. **El jugador quiere que cuando su personaje salte, suene un sonido de efecto para así hacerlo más entretenido.**

Figura 3.3: Ejemplo de requisitos del juego de plataformas que entran en la categoría *Could*



REQUISITOS JUEGO 2

MUST HAVE

1. **El jugador quiere que el juego tenga un fondo base para que sea más vistoso.**

   Cuando el jugador acceda al juego, si el fondo ha cargado correctamente, entonces se mostrará el fondo correspondiente.

   Cuando el jugador acceda al juego, si el fondo no ha cargado, entonces se mostrará un círculo de carga.

2. **El jugador quiere que aparezca el listado de niveles del juego para así poder saber cuáles están disponibles y cuáles están bloqueados.**

   Cuando el jugador vaya a entrar en el juego, si ha accedido a este, entonces aparecerá un listado de los niveles del juego.

   Cuando el jugador vaya a entrar en el juego, si no ha accedido a este, entonces no pasará nada.

3. **El jugador quiere que se bloqueen los niveles a los que no puede acceder todavía, para que así se restrinja su acceso a los niveles.**

   Cuando el jugador esté en el juego, si ha accedido al listado de niveles, entonces se bloquearán los niveles a los que no puede acceder todavía.

4. **El jugador quiere que aparezcan con un icono de candado los niveles del juego que estén bloqueados para así saber a cuáles no puede acceder.**

   Cuando el jugador esté en el juego, si ha accedido al listado de niveles y hay alguno bloqueado, entonces se mostrarán con el icono de un candado.

   Cuando el jugador esté en el juego, si no hay ningún nivel bloqueado, entonces ninguno se mostrará con el icono del candado.

5. **El jugador quiere que aparezca el número de cada nivel del juego que esté desbloqueado para así saber a cuáles puede acceder.**

   Cuando el jugador esté en el juego, si ha accedido al listado de niveles y las imágenes

6. **El jugador quiere que cuando se complete un nivel, se desbloquee el siguiente, para así poder continuar el juego.**

   Cuando el jugador esté en el juego, si ha superado un nivel, entonces se desbloqueará el siguiente nivel.

   Cuando el jugador esté en el juego, si no ha superado un nivel, entonces no se desbloqueará el siguiente nivel.

7. **El jugador quiere poder acceder un nivel pulsando uno de los disponibles en el listado de niveles para así poder jugar.**

   Cuando el jugador esté en el juego, si está en el listado de niveles, y pulsa en uno de los niveles disponibles, entonces aparecerá la pantalla de dicho nivel.

   Cuando el jugador esté en el juego, si está en el listado de niveles y pulsa uno de los niveles que no están disponibles aún, entonces aparecerá un mensaje de "¡Este nivel está bloqueado!"

8. **El jugador quiere que cada nivel se divida en diez subniveles para que se alarguen más.**
   Cuando el jugador esté en el juego, si ha accedido a un nivel, entonces este tendrá 10 subniveles más.

9. **El jugador quiere que cada nivel sea más difícil que el anterior para así hacer el juego más entretenido.**

   Cuando el jugador esté en el juego, si ha superado un nivel, quiere que el siguiente tenga fotos de comida cuyas raciones de carbohidratos sean más complicadas de identificar o más específicas para aumentar la dificultad del juego.

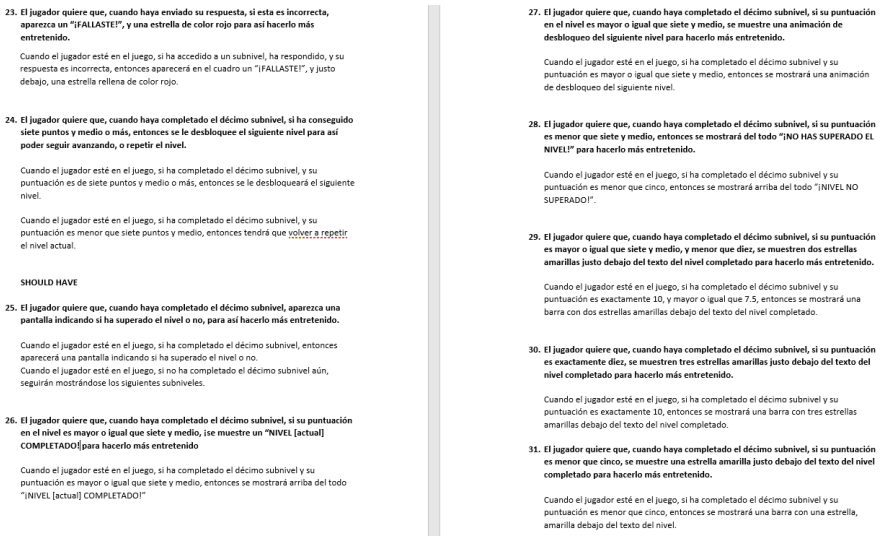
   Cuando el jugador esté en el juego, si no ha superado un nivel, quiere que la dificultad de dicho nivel se mantenga.

10. **El jugador quiere que cada vez que se acceda a un nivel, los subniveles aparezcan de forma aleatoria para hacer el juego más entretenido.**

    Cuando el jugador esté en el juego, al acceder a un nivel, los subniveles saldrán de forma aleatoria y sin que se repitan.

11. **El jugador quiere que en cada subnivel aparezca una imagen de comida, para así tener que identificarla y poder jugar.**

Figura 3.4: Ejemplo de requisitos del juego de preguntas que entran en la categoría *Must*



23. **El jugador quiere que, cuando haya enviado su respuesta, si esta es incorrecta, aparezca un "¡FALLASTE!", y una estrella de color rojo para así hacerlo más entretenido.**

    Cuando el jugador esté en el juego, si ha accedido a un subnivel, ha respondido, y su respuesta es incorrecta, entonces aparecerá en el cuadro un "¡FALLASTE!", y justo debajo, una estrella rellena de color rojo.

24. **El jugador quiere que, cuando haya completado el décimo subnivel, si ha conseguido siete puntos y medio o más, entonces se le desbloquee el siguiente nivel para así poder seguir avanzando, o repetir el nivel.**

    Cuando el jugador esté en el juego, si ha completado el décimo subnivel, y su puntuación es de siete puntos y medio o más, entonces se le desbloqueará el siguiente nivel.

    Cuando el jugador esté en el juego, si ha completado el décimo subnivel, y su puntuación es menor que siete puntos y medio, entonces tendrá que volver a repetir el nivel actual.

SHOULD HAVE

25. **El jugador quiere que, cuando haya completado el décimo subnivel, aparezca una pantalla indicando si ha superado el nivel o no, para así hacerlo más entretenido.**

    Cuando el jugador esté en el juego, si ha completado el décimo subnivel, entonces aparecerá una pantalla indicando si ha superado el nivel o no.
    Cuando el jugador esté en el juego, si no ha completado el décimo subnivel aún, seguirán mostrándose los siguientes subniveles.

26. **El jugador quiere que, cuando haya completado el décimo subnivel, si su puntuación en el nivel es mayor o igual que siete y medio, ¡se muestre un "NIVEL [actual] COMPLETADO!¡para hacerlo más entretenido**

    Cuando el jugador esté en el juego, si ha completado el décimo subnivel y su puntuación es mayor o igual que siete y medio, entonces se mostrará arriba del todo "¡NIVEL [actual] COMPLETADO!"

27. **El jugador quiere que, cuando haya completado el décimo subnivel, si su puntuación en el nivel es mayor o igual que siete y medio, se muestre una animación de desbloqueo del siguiente nivel para hacerlo más entretenido.**

    Cuando el jugador esté en el juego, si ha completado el décimo subnivel y su puntuación es mayor o igual que siete y medio, entonces se mostrará una animación de desbloqueo del siguiente nivel.

28. **El jugador quiere que, cuando haya completado el décimo subnivel, si su puntuación es menor que siete y medio, entonces se mostrará del todo "¡NO HAS SUPERADO EL NIVEL!" para hacerlo más entretenido.**

    Cuando el jugador esté en el juego, si ha completado el décimo subnivel y su puntuación es menor que cinco, entonces se mostrará arriba del todo "¡NIVEL NO SUPERADO!".

29. **El jugador quiere que, cuando haya completado el décimo subnivel, si su puntuación es mayor o igual que siete y medio, y menor que diez, se muestren dos estrellas amarillas justo debajo del texto del nivel completado para hacerlo más entretenido.**

    Cuando el jugador esté en el juego, si ha completado el décimo subnivel y su puntuación es exactamente 10, y mayor o igual que 7.5, entonces se mostrará una barra con dos estrellas amarillas debajo del texto del nivel completado.

30. **El jugador quiere que, cuando haya completado el décimo subnivel, si su puntuación es exactamente diez, se muestren tres estrellas amarillas justo debajo del texto del nivel completado para hacerlo más entretenido.**

    Cuando el jugador esté en el juego, si ha completado el décimo subnivel y su puntuación es exactamente 10, entonces se mostrará una barra con tres estrellas amarillas debajo del texto del nivel completado.

31. **El jugador quiere que, cuando haya completado el décimo subnivel, si su puntuación es menor que cinco, se muestre una estrella amarilla justo debajo del texto del nivel completado para hacerlo más entretenido.**

    Cuando el jugador esté en el juego, si ha completado el décimo subnivel y su puntuación es menor que cinco, entonces se mostrará una barra con una estrella, amarilla debajo del texto del nivel.

Figura 3.5: Ejemplo de requisitos del juego de preguntas que entran en la categoría *Should*

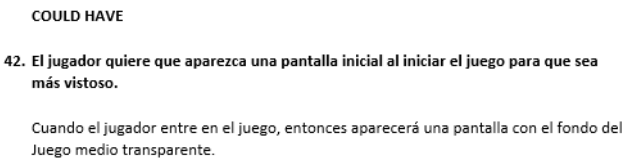


Figura 3.6: Ejemplo de un requisito del juego de preguntas que entra en la categoría *Could*

El resultado de aplicar la técnica mencionada se puede observar en las Imágenes [número de las imágenes]. Cada Historia de Usuario recogía tanto el *Happy path*, es decir, la situación en la que usuario realiza el caso de uso sin errores durante el proceso, como los *Corner cases*, que es cuando hay resultados inesperados debido a que no se ha realizado lo esperado durante el caso de uso.

Cabe mencionar que la definición de la Historia de Usuario la realizamos en negrita. Posteriormente, se definieron los *Corner cases*, seguidos del *Happy path*. Podemos observar la estructura de cada Historia de Usuario en las figuras 3.1, 3.2 3.3, 3.4, 3.5, y 3.6.

3.2. Herramientas utilizadas

3.2.1. Herramientas para el diseño de software

Tras la captura de requisitos, pasamos a la elección de herramientas que usaríamos para este proyecto. Puesto que la fase de diseño es previa a la de desarrollo, lo primero que hicimos fue elegir las herramientas -si fueran necesarias- para hacer tanto diagramas como Mockups.

En nuestro caso, dado que los diagramas iniciales los hicimos a mano sobre papel, solo escogimos la herramienta necesaria para hacer mockups que más tarde nos servirían para hacer la interfaz del juego. En este caso, dicha herramienta fue Balsamiq Mockups, debido a que era bastante intuitiva de usar, y tenía numerosos elementos prefabricados para la interfaz, como se puede observar en la figura 3.7 lo cual nos facilitaría bastante realizar los mockups.

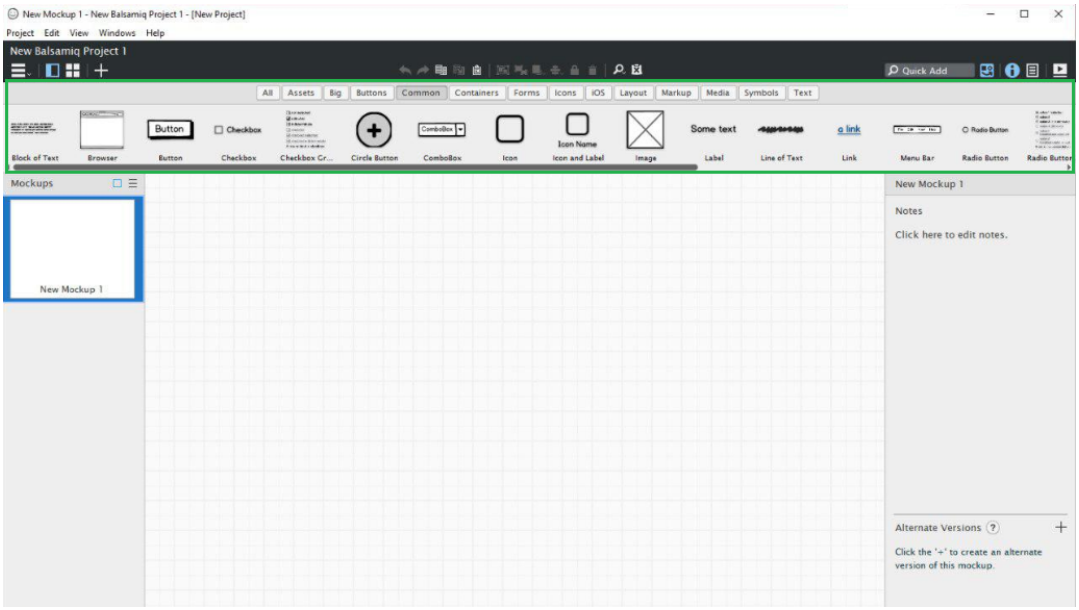


Figura 3.7: Captura de Balsamiq Mockups

3.2.2. Herramientas para el desarrollo de software

Debido a la gran cantidad de documentación que hay en internet y la versatilidad que ofrece, decidimos usar Unity como motor gráfico. Usamos las versiones actualizadas de 2018 (específicamente Unity 2018.4.4f1), incluyendo actualizaciones de la misma. Por las mismas razones decidimos programar en C#. Para instalar y acceder a Unity de manera sencilla, nos instalamos Unity Hub, una herramienta que ayuda a gestionar versiones y proyectos de Unity.

En cuanto al entorno de desarrollo que usar, probamos tanto Monobehaviour como Visual Studio, pero debido a que Monobehaviour no se puede usar en versiones de Unity superiores a la versión de 2017, escogimos Visual Studio, (utilizando las versiones 2017 y 2019).

Para generar nuestra base de datos, utilizamos Visual Studio Code, puesto que nos parecía más *user friendly* y más simple que Visual Studio. Además, se pueden descargar fácilmente plugins para utilizar cualquier lenguaje. Dicha versatilidad se ajustó a nuestras necesidades, ya que no sabíamos en qué formato estaría nuestra base de datos, y pudimos hacer pruebas con C# también.

Previendo que uno de nosotros pudiera sobreescribir el trabajo del otro, y, para ir desarrollando cada Historia de Usuario en orden, decidimos trabajar en un repositorio privado de Trello para gestionar el reparto de tareas. Teníamos distribuido el tablero de Trello, como se observa en la figura 3.8 de la siguiente manera:

- había una columna “TO DO”, que es donde se guardaban las tareas pendientes de hacer.

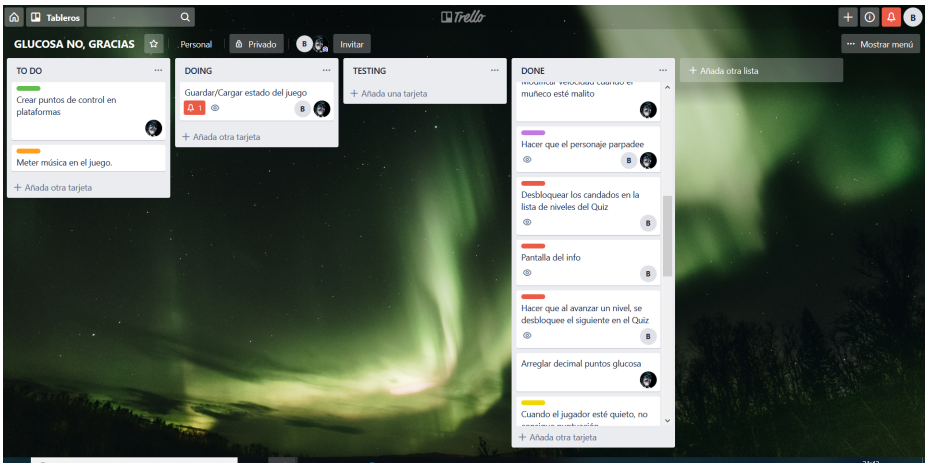


Figura 3.8: Tablero de Trello que usamos para distribuir las tareas de este proyecto

- Le seguía la columna "DOING", usada para las tareas que se estaban realizando en ese momento.
- Después se encontraba la columna "TESTING", en la cual se probaban las tareas que se habían realizado previamente.
- Una vez pasaran todas las pruebas las tareas de "TESTING", pasaban a la columa de "DONE", donde ya se daban por finalizadas.

De esta forma, pudimos trabajar en paralelo y supervisar lo que hacía cada uno. Gracias a ésto también pudimos ver el avance lento (pero constante) de este proyecto.

Por último, nos quedaba por decidir el software que utilizaríamos para crear y editar los sprites del videojuego. Dudamos entre Photoshop o Gimp, y lo que hizo inclinar la balanza fue que este último es gratuito. La versión de Gimp que usamos es la 2.8.10.

3.2.3. Sistema de Control de Versiones Distribuido

Para organizar todos los archivos relacionados con el proyecto, utilizamos un sistema de control de versiones distribuido, ya que estos permiten recuperar versiones anteriores, así como trabajar en paralelo sin sobreescribir el trabajo de otras personas. Concretamente, utilizamos Github, tanto la versión de escritorio llamada Github Desktop, como la versión que se utiliza desde la consola de comandos, git.

Como se puede ver en la figura 3.9, nuestro repositorio de Github -que está creado como un repositorio privado- está organizado por carpetas de la siguiente forma:

- Por un lado, tenemos la carpeta de "Assets", que es donde se han ido guardando todos los sprites que se han ido creando u obtenido de internet. Está dividida

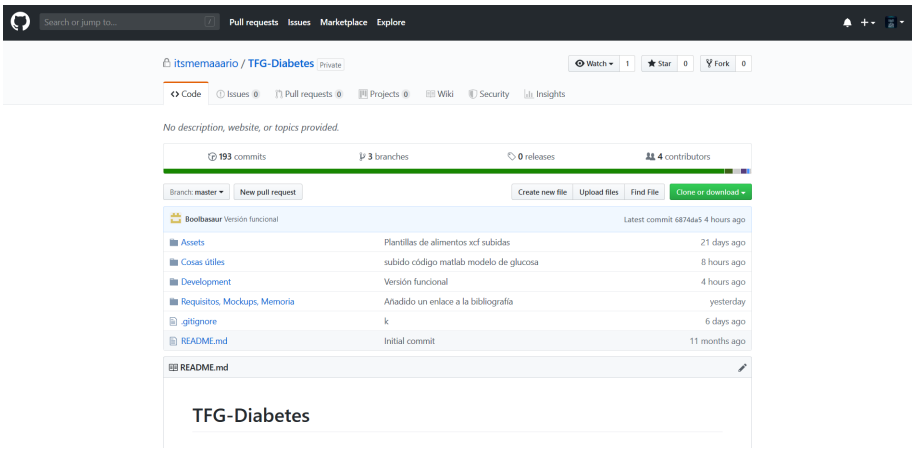


Figura 3.9: Distribución del repositorio del proyecto en Github

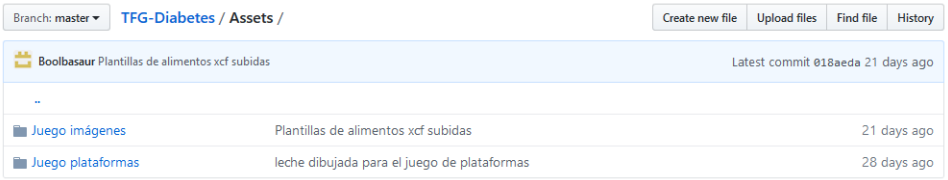


Figura 3.10: Distribución de la carpeta “Assets” en Github

a su vez en dos carpetas más, una para el juego de plataformas, “Juego plataformas” y otra para el juego de preguntas, “Juego imágenes”, como se puede observar en la figura 3.10.

- Le sigue la carpeta “Cosas útiles”, que es una carpeta de carácter más general hemos utilizado para guardar archivos que contengan información sobre la diabetes, así como todas las fuentes que hemos consultado a lo largo de nuestro proyecto, ya sean tanto libros en formato pdf, o enlaces a fuentes que están online.

- En “Development”, usada para guardar e ir actualizando el código, se alojan tres carpetas más, tal y como vemos en la figura 3.11:

  - “Juego principal”, que es donde se encuentra la versión final del videojuego como proyecto de Unity.
  - “Prototipo”, donde se ha realizado el prototipo del juego de plataformas.
  - Y por último, tenemos “Quiz”, que es donde se encuentra el prototipo del juego de preguntas.

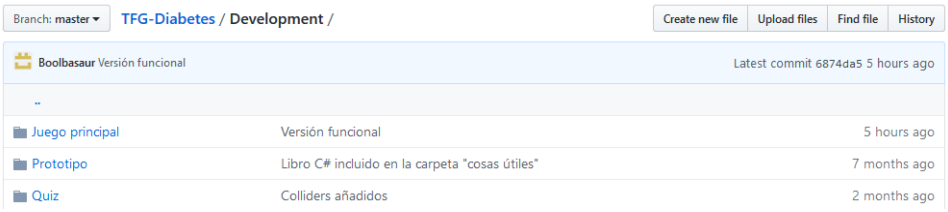


Figura 3.11: Distribución de la carpeta “Development” en Github

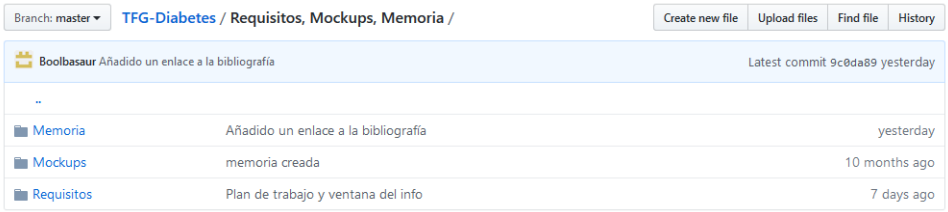


Figura 3.12: Distribución de la carpeta “Requisitos, Mockups, y Memoria” en Github

- Tenemos también una carpeta llamada “Requisitos, Mockups, y Memoria”, utilizada para toda la parte del diseño de software y la documentación necesaria para la realización de la memoria. Se subdivide, a su vez, como se ve en la figura 3.12, en tres carpetas más:
  - “Memoria”, utilizada para guardar toda la documentación reunida para la realización de la memoria.
  - “Mockups”, donde se guardaron los Mockups realizados para los prototipos.
  - “Requisitos”, que contenía las Historias de Usuario que escribimos para cada juego.
- Añadimos además un “.gitignore” para evitar que cada vez que subíamos algún fragmento nuevo de código al respositorio cargara archivos innecesarios de Unity.

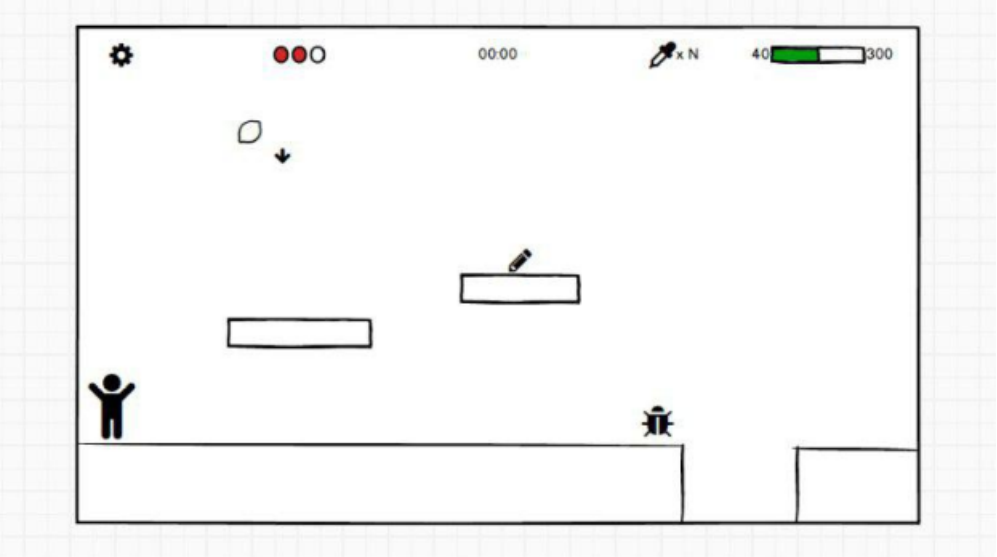


Figura 3.13: Mock de un nivel básico en el juego de plataformas

3.3. Diseño de mockups y diagramas iniciales

3.3.1. Diseño de Mockups

En la tercera fase del diseño del videojuego teníamos un objetivo claro: establecer un esquema básico del videojuego del que partir, y asegurarnos de tener claras cada una de las Historias de Usuario definidas.

Para esto, utilizando Balsamiq Mockups, hicimos un boceto básico de cada uno de los juegos. Como se puede observar, en la figura 3.13 aparece lo que sería un nivel básico del juego de plataformas, con elementos como las vidas, el botón de configuración, el botón para administrarse la insulina, el jugador, etc. También vemos en la figura 3.14 lo que sería la barra de glucosa del jugador, y cómo variaría en función de los valores de glucosa en sangre que tenga este.

Por otra parte, en el mock del quiz, tal y como se observa en la figura 3.15, se ve una estructura clara de cada uno de los subniveles (es decir, cada pregunta dentro de un nivel) del juego. Tenemos arriba lo que sería la pregunta sobre gramos de carbohidratos, y debajo a la izquierda la imagen del alimento o la comida, con las posibles respuestas justo al lado. A su vez, se puede apreciar abajo una flecha para volver al menú principal del juego de preguntas, así como un botón de configuración.

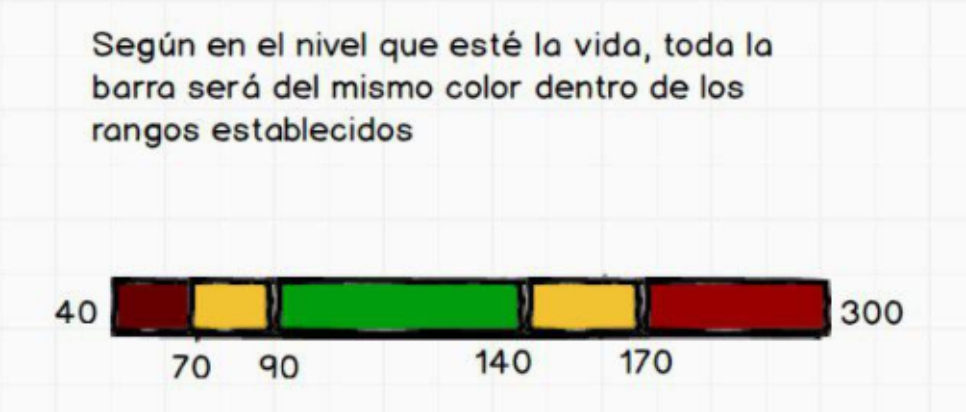


Figura 3.14: Mock que ilustra la barra de glucosa y los colores que tiene en función de sus valores

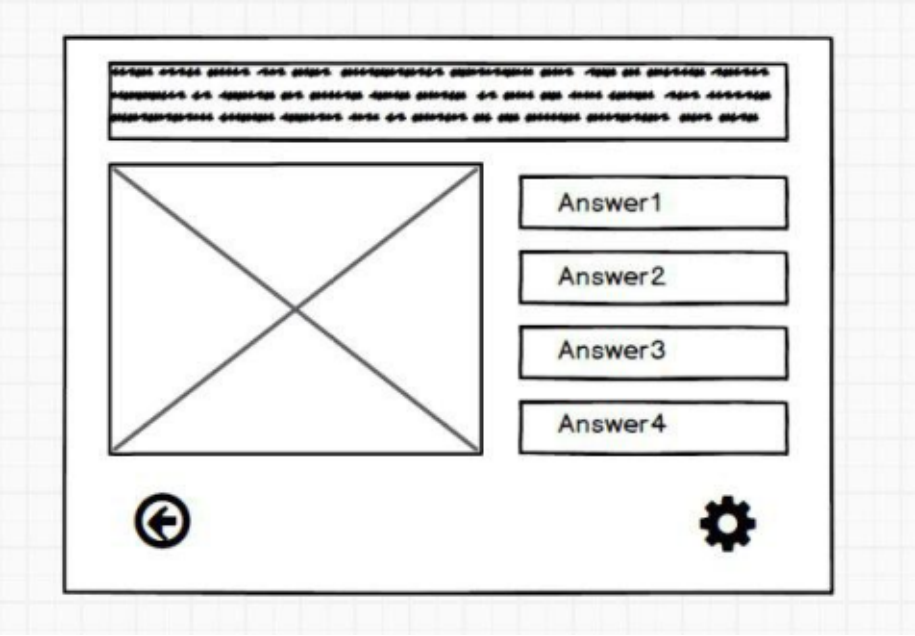


Figura 3.15: Mock de un subnivel básico del quiz

3.3.2. Diagramas iniciales

En las figuras 3.16 y 3.17 se puede observar el diagrama de cada juego que ideamos para tener una idea sobre el diseño del software. Apenas tiene elementos, puesto que nos basamos en las nociones de Ingeniería de Software, pero todavía no nos desenvolvíamos perfectamente con Unity y C#. Estos diagramas serían entonces una base de la que partir, lo que dio pie a que empezáramos a desarrollar código.

Se puede apreciar en la figura 3.16 que el diagrama del quiz tiene menos elementos debido a que no teníamos demasiado claro la estructura que podría tener un juego de preguntas, a diferencia del juego de plataformas, del cual sí teníamos algunas ideas previas.

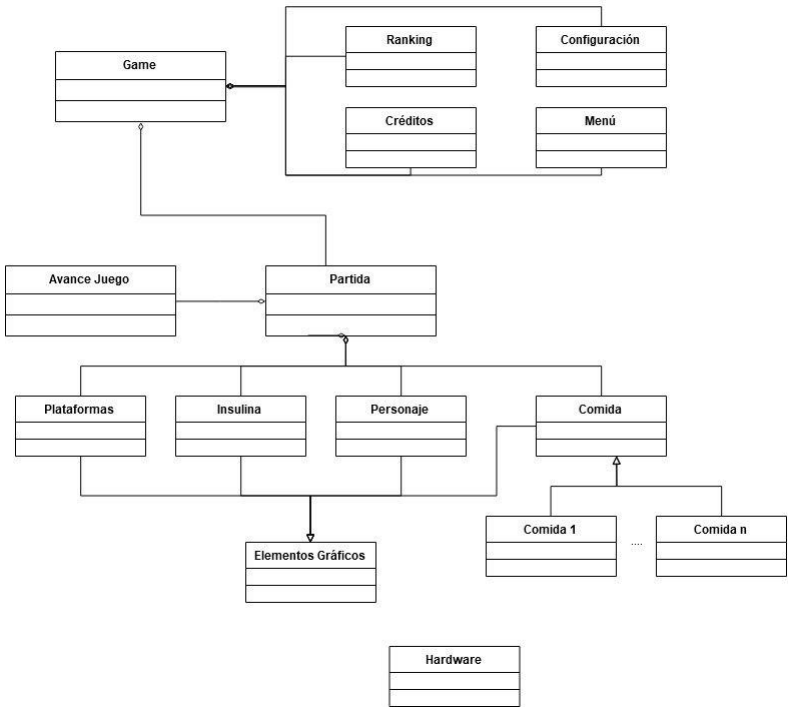


Figura 3.16: Diagrama de clases del juego de plataformas

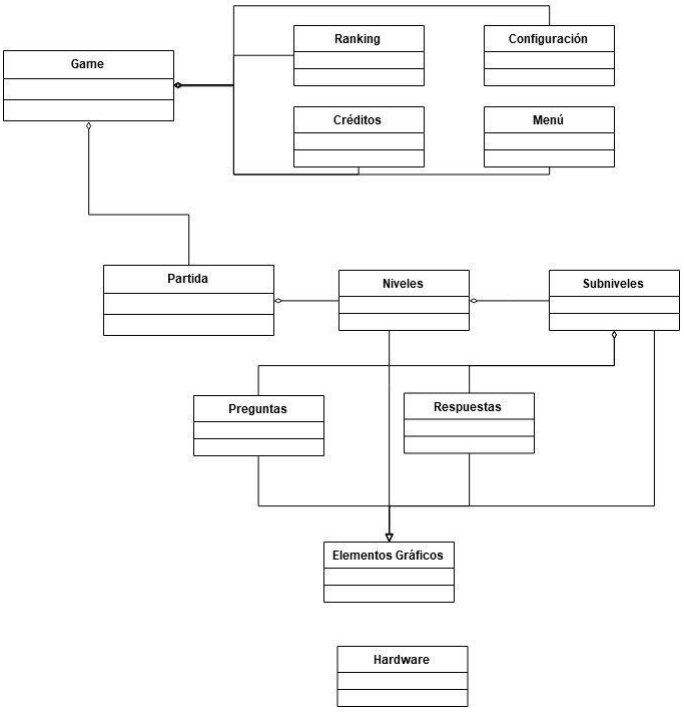


Figura 3.17: Diagrama de clases del juego de preguntas

Capítulo 4

Desarrollo del software

Greatness, from small beginnings

Uncharted

En este capítulo explicamos cómo se ha ido programando el código desde la creación de prototipos de prueba para poner en práctica nuestros conocimientos sobre Unity y C#, hasta la mejora de la Interfaz de Usuario de la versión final del videojuego.

4.1. Creación de prototipos

4.1.1. Prototipo del juego de plataformas

Una vez terminada la fase de diseño, ya habíamos obtenido nociones básicas sobre Unity y C#, puesto que estuvimos formándonos de forma paralela a la elaboración del diseño del software(12), (13), (14), y (15). Utilizamos dichas nociones para empezar a programar funcionalidades muy básicas sobre lo que serían, posteriormente, los prototipos de cada juego.

Comenzamos con la elaboración de varios prototipos del juego de plataformas, ya que es el juego que más complejidad creíamos que podría tener. Cada uno hicimos diversos prototipos que solo tenían en común los elementos que incluían: un jugador que se movía por varias plataformas existentes, como se muestra en las figuras 4.1, 4.2 y 4.3. Si el jugador se caía de las plataformas, reaparecía de nuevo en la misma posición en la que estaba al inicio de la ejecución del prototipo.

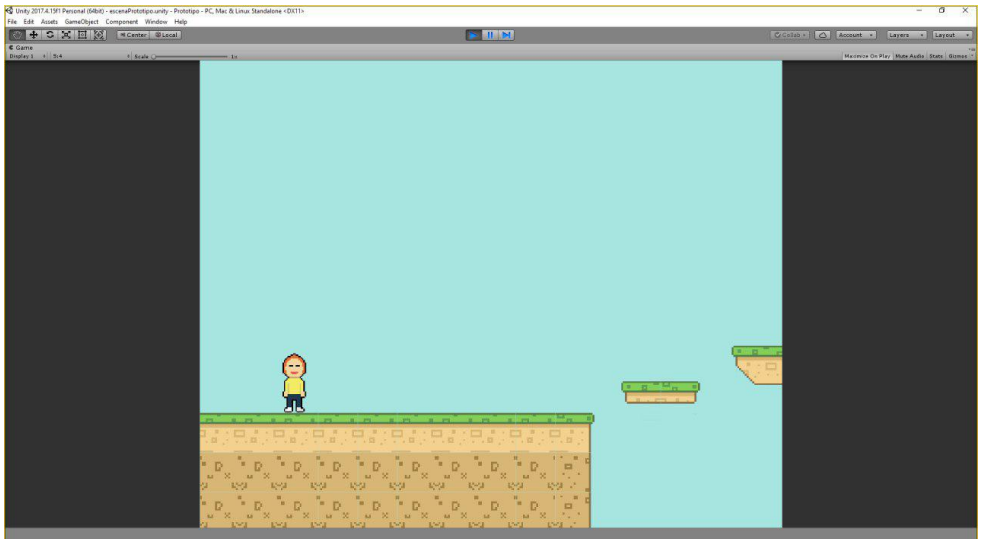


Figura 4.1: Imagen donde se muestra el prototipo del juego de plataformas.

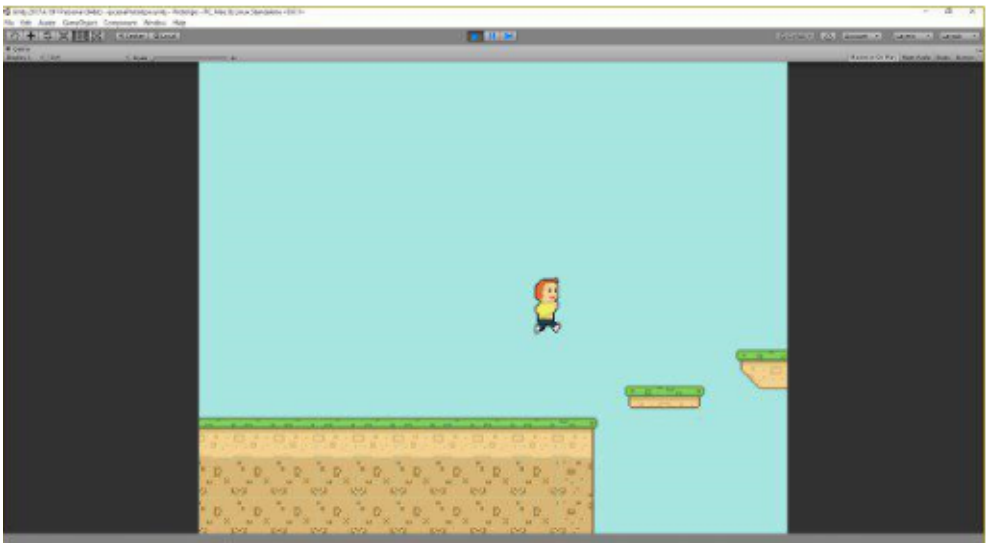


Figura 4.2: Imagen del prototipo del juego de plataformas en el que aparece el personaje saltando de una plataforma a otra.

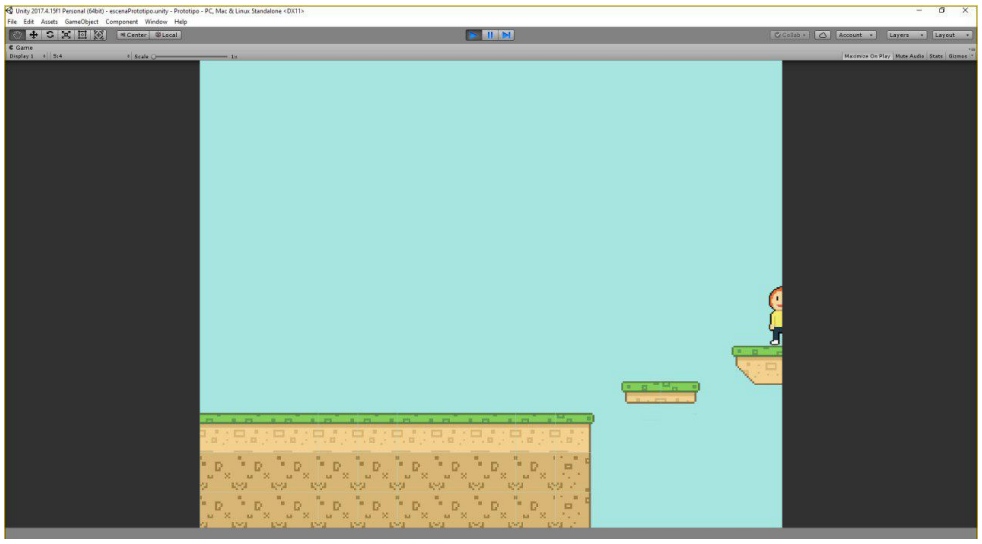


Figura 4.3: Imagen del prototipo del juego de plataformas en el que se observa que no se puede avanzar.

Lo que más nos preocupaba era el movimiento y la colisión del jugador; cómo pasaba de estar parado a estar en movimiento, o incluso saltando. Tras manejar este aspecto, acordamos presentar uno de los prototipos al tutor de este trabajo de fin de grado para que le diera el visto bueno.

4.1.2. Prototipo del juego de preguntas

En cuanto al prototipo del juego de preguntas, aunque fuera más fácil que el del juego de plataformas, puesto que no tenía cosas tan complejas como las físicas del personaje, nos resultó más arduo desarrollarlo. No estaba tan bien documentado como podía estarlo un juego de plataformas en 2D, y teníamos nociones muy básicas de Unity y C# hasta la fecha, por lo que tuvimos que hacer diversas pruebas hasta desarrollar un prototipo funcional que nos sirviera como base para el juego principal.

El resultado del prototipo final de preguntas, fue un juego básico de preguntas, con imágenes obtenidas de internet. Si acertabas la pregunta, se mostraba la respuesta elegida en verde y pasaba a la siguiente pregunta, como podemos ver en la figura 4.4. Si por el contrario, fallabas la pregunta, se mostraba el botón de la respuesta rojo, y pasaba a la siguiente pregunta también, como se observa en la figura 4.5. Cuando terminaba la secuencia de preguntas, se mostraba en los logs de la consola de Unity la puntuación final.

Cabe mencionar que, excepto la respuesta correcta, las demás se generaban aleatoriamente para cada pregunta en función del valor de la correcta. Si el valor de la correcta era 100, se generaban 3 respuestas más aleatorias cuyos resultados no se alejan más de cierta cantidad que establecimos en el código.

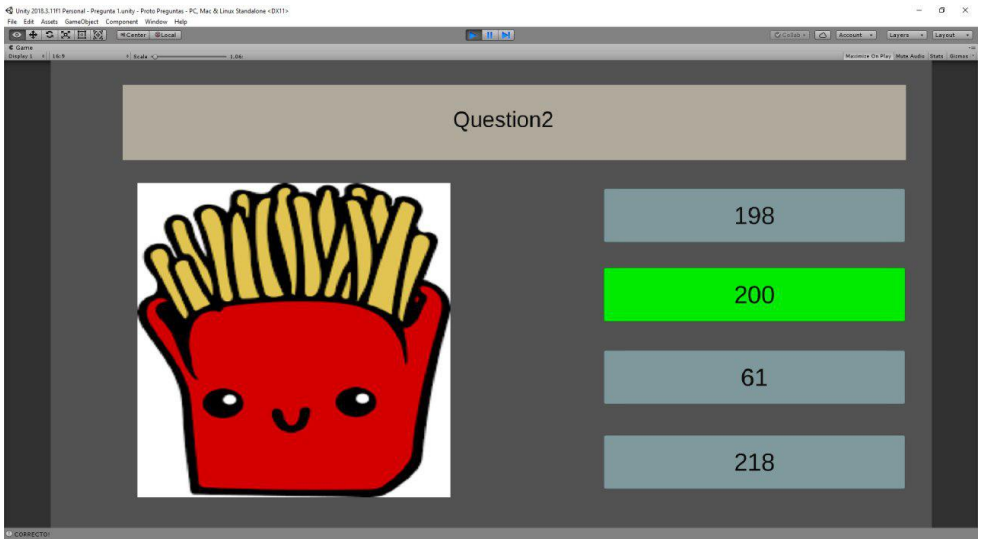


Figura 4.4: Imagen donde se muestra una pregunta acertada en el prototipo del juego de preguntas.

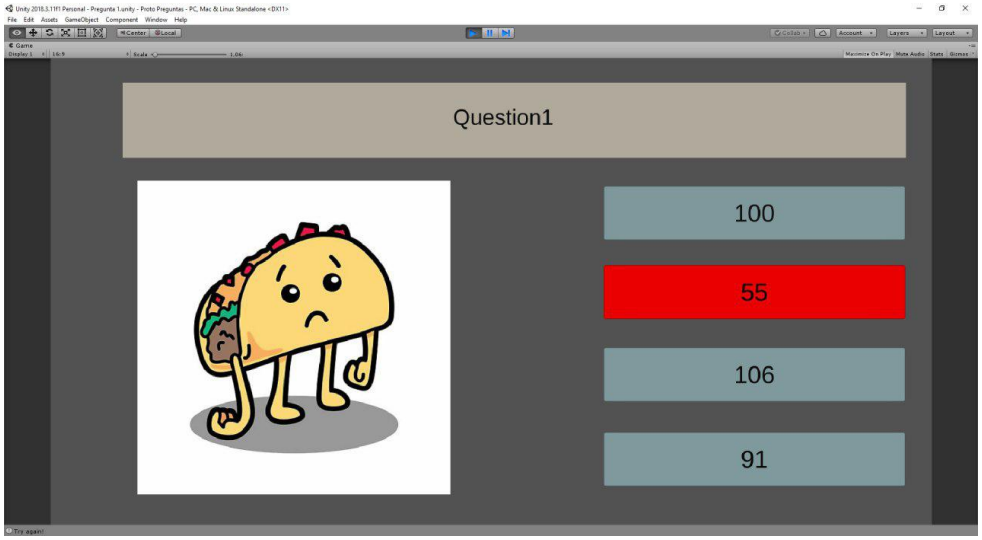


Figura 4.5: Imagen donde se muestra una pregunta fallada en el prototipo del juego de preguntas

4.2. Unificación de prototipos

Tras haber terminado el desarrollo de cada prototipo por separado, iniciamos el proceso de unificación de cada prototipo para tener la base para el videojuego

Figura 4.6: Imagen que muestra el menú principal inicial

definitivo. Lo primero que hicimos fue crear un menú principal desde el que podríamos acceder a cada juego, el cual se puede observar en la figura 4.6. Posteriormente, añadimos el prototipo de plataformas y comprobamos que funcionaba correctamente accediendo a éste desde el menú principal recientemente creado.

Creíamos que añadir el otro prototipo supondría la misma dificultad que con el juego de plataformas. No obstante, la versión en la que hicimos el prototipo de Unity nos dio problemas debido a que, al no ser estable, la acabaron quitando de las versiones que se podían descargar de Unity Hub. Tuvimos que rehacer el juego de preguntas sobre otra versión más nueva que sí era LTS (*Long Term Support*), es decir, estable.

Una vez rehecho el prototipo de preguntas, lo añadimos al proyecto que ya habíamos montado con el menú principal y el otro prototipo. Solo tuvimos que añadir el acceso desde el menú creado, y ya teníamos ambos juegos dentro de un proyecto común sobre el cual seguir trabajando. Dicho proyecto sería posteriormente la versión final del videojuego.

Como añadido, en vez de acceder a cada juego directamente desde el menú principal, creamos menús intermedios de cada juego particular, desde el que poder acceder al menú de opciones, al juego, o volver al menú principal. El menú del juego de plataformas se puede observar en la figura 4.7, y el del juego de preguntas en la figura 4.8.

Figura 4.7: Imagen que muestra el menú inicial del juego de plataformas

Figura 4.8: Imagen que muestra el menú inicial del juego de preguntas

4.3. Desarrollo de la versión final del videojuego

Con los juegos ya unificados, solo tuvimos que ir desarrollando en paralelo cada uno de éstos. Fuimos siguiendo una serie de objetivos que marcamos a través de las tareas que había distribuidas en Trello. Intentamos que el avance de ambos juegos fuera progresivo, sin que hubiera un gran brecha entre ellos, puesto que no queríamos que uno estuviera sin apenas funcionalidades, y el otro casi terminado.

4.3.1. Desarrollo del juego de plataformas

Siguiendo las prioridades definidas en las Historias de usuario del juego de plataformas, comenzamos el desarrollo del juego de plataformas con las funcionalidades básicas del personaje. De primeras solo añadimos elementos gráficos: un corazón y texto para las vidas, un alimento de prueba, una barra de glucosa, el texto para la puntuación, y la insulina que podía administrarse el jugador junto con el texto que mostraba la cantidad. Posteriormente, implementamos la funcionalidad de cada elemento. Las vidas, que fue lo primero que conseguimos que funcionara, inicialmente solo se restaban cuando el personaje se caía de las plataformas y moría, lo cual hacía que a éste se le restara una vida. Después de realizar la implementación de las vidas, creamos los scripts correspondientes de los consumibles del personaje para poder probar que los niveles de glucosa subieran adecuadamente y en un tiempo determinado. Dichos scripts contendrían posteriormente el cálculo lineal de la insulina prandial. Testeamos a través de la consola de Unity que el personaje pudiera comer correctamente creando un alimento de prueba con valores aleatorios, y cuando los resultados fueron los esperados, implementamos la funcionalidad de la barra de glucosa.

La funcionalidad de la barra de glucosa consistía actualizar el sprite de dicha barra en función de los niveles de glucosa del jugador (niveles que se tienen que encontrar comprendidos entre 0 y 400, puesto que es lo que abarca la barra por completo). Además, controlamos que la barra de glucosa cambiara de color en función de los niveles de ésta: si la glucosa del personaje está por debajo de 40 o por encima de 300, dicha barra se mostrará de color rojo, como se observa en la figura 4.9. Si la glucosa está por debajo de 70 o por encima de 140, la barra de glucosa se verá de color naranja, como se ve en la figura 4.10. Si los niveles de glucosa del personaje son adecuados, es decir, que éste tenga la glucosa entre 70 y 140, la barra se mostrará verde, como observamos en la figura 4.11. Cuando los niveles de glucosa estaban por encima de 300 o por debajo de 40 durante más de 10 segundos, el personaje moría, y volvía al inicio del nivel con una vida menos.

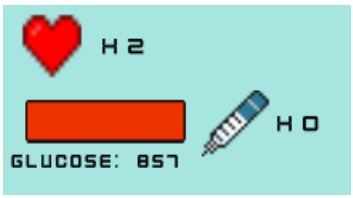


Figura 4.9: Imagen que muestra la barra de glucosa del personaje con valores excesivamente elevados

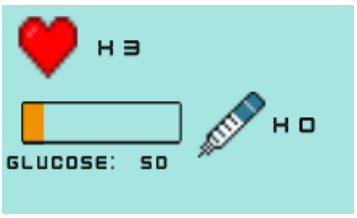


Figura 4.10: Imagen donde se puede ver la barra de glucosa del personaje con niveles de glucosa elevados

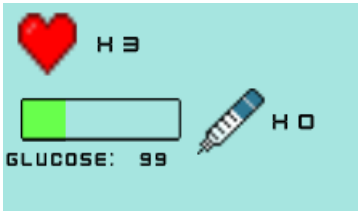


Figura 4.11: Imagen que muestra el menú inicial del juego de preguntas

Habiendo completado la funcionalidad de la barra de glucosa del personaje, incorporamos también el consumible de la insulina, ya que reutilizaríamos scripts previamente realizados para los alimentos. Como mencionamos en el párrafo anterior, al principio solo añadimos el elemento. No obstante, posteriormente le añadimos la acción de botón, ya que el jugador tendría que pulsar la insulina que se muestra al lado de la barra de glucosa para administrar una unidad de insulina, como se puede ver en las figuras 4.9, 4.10, y 4.11. Al pulsar en la insulina, los niveles de glucosa en sangre del personaje bajan a lo largo de un minuto (puesto que un minuto de juego es una hora en la vida real). De este modo, simulamos la administración de una dosis de insulina prandial, teniendo en cuenta las gramos de carbohidratos que ha ingerido el jugador, y el tiempo de absorción de dichos carbohidratos (16).

Lo siguiente que incorporamos al juego fue la puntuación que conseguía el personaje en el nivel. Desde que comenzaba el juego hasta que el personaje perdía una vida, la puntuación aumentaba. Cada vez que el jugador reaparecía al principio del nivel -bien porque moría, o porque empezaba el juego- la puntuación se reiniciaba. Una vez implementada la funcionalidad en el código, solo tuvimos que actualizar el texto en la pantalla del nivel. Además, añadimos una escena de ¡Perdiste!, como se observa en la figura 4.12, que aparecía cuando el personaje perdía todas las vidas. De esta manera, el usuario podía tener interacción por parte del juego cuando perdiese.

Figura 4.12: Imagen que muestra la escena de "¡Perdiste!" del juego de plataformas

Después hicimos todos los *prefabs* necesarios para tener todos los alimentos que el jugador podía consumir creados, como se puede ver en la imagen 4.13. Los *prefabs* son objetos con ciertos componentes añadidos que puedes crear a tu gusto en Unity y guardar en la carpeta de *Assets*. La ventaja de los *prefabs* es que puedes generar muchos objetos iguales o con características similares de manera sencilla. Para cada alimento tuvimos en cuenta cuántas gramos de carbohidratos tenían, y su velocidad de absorción, exactamente de la misma forma que con la insulina (17) (16).

Figura 4.13: Imagen que muestra los prefabs que creamos para cada alimento

Posteriormente añadimos la funcionalidad del botón del menú de opciones. Este menú ya estaba creado, y se podía acceder desde el menú del juego de plataformas, pero no desde el propio juego. Lo único que tuvimos que hacer fue implementar el elemento de la configuración del juego como un botón, y que cuando se pulsara, se mostrara el menú de configuración por delante del nivel, a la vez que se congelaba todo el juego. Si se salía del menú pulsando en la "x", se retomaba el juego por donde

se había quedado y se ocultaba el menú. Como se puede ver en la figura 4.14, en este juego ofrece la opción de subir o bajar el volumen, para cuando éste tenga música incorporada.

Figura 4.14: Imagen donde se ve el menú de configuración del juego de plataformas

Teniendo el juego básico casi completo, nos centramos en detalles menos prioritarios. Conseguimos que, de las puntuaciones que se podían conseguir en el nivel (tantas como vidas iniciales tenía el jugador), se mostrara la más alta en el menú del "¡Perdiste!". Además, ya no se obtenía puntuación simplemente estando en el nivel: ahora el personaje tenía que estar moviéndose para que siguiera subiendo la puntuación, lo cual es el equivalente a hacer ejercicio.

También hicimos que se generara la comida e insulina aleatoriamente a lo largo del nivel. Para esto, decidimos que se generaría solo desde el punto máximo alcanzado por el personaje en el nivel hacia delante, para así evitar que el jugador se quede al principio. Cabe mencionar que es más probable que se genere insulina a que se genere cualquier alimento, debido a que esto disminuye la curva de dificultad del juego. Otro añadido más bien visual fue que el personaje parpadeara cuando tenía los niveles de glucosa por debajo de 40 o por encima de 300.

Aprovechando que teníamos un script para generar comida aleatoriamente, también generamos las nubes del nivel aleatoriamente para hacer que el escenario del nivel fuera aún más entretenido, y aprovechamos para ampliar el nivel para así aumentar el tiempo de juego de los usuarios. Otra implementación que añadimos en la etapa final del desarrollo de este juego, fue que cuando el jugador se quedara sin vidas, se le obligara a jugar al juego de preguntas para así poder conseguir más. Así, conseguiríamos que el jugador pudiera probar ambos juegos, asegurando que éste viera un progreso en los dos, puesto que también guardamos la partida en un archivo de Unity. La partida la guardamos a través de variables especificas de los session de cada juego, que cargan al inicio las variables y las van guardando según se progrese en cada uno de éstos. Finalmente, como elemento extra, añadimos una estrella que

daba puntos extra al jugador, en el caso de cogerla. De este modo, se pueden hacer caminos de mayor dificultad a lo largo del nivel y forzar al jugador a pasar por éstos si quieren conseguir una puntuación mejor. También creamos un *ranking* para que el jugador pudiera ver las puntuaciones obtenidas en el juego, al cual se accede desde el menú del juego de plataformas. Sin embargo, solo se muestran las 5 mejores puntuaciones obtenidas, como se puede observar en la figura 4.15. En el caso en el que no haya suficientes puntuaciones guardadas, se mostrará "- - -" en la posición en la que falten, como también se ve en la figura 4.15. De este modo, el nivel del juego de plataformas quedaría como se observa en la figura 4.16.

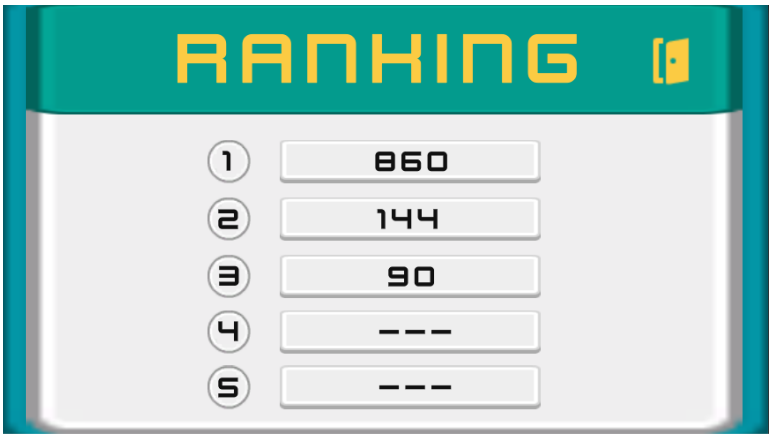


Figura 4.15: Imagen donde se puede ver el *ranking* del juego de plataformas

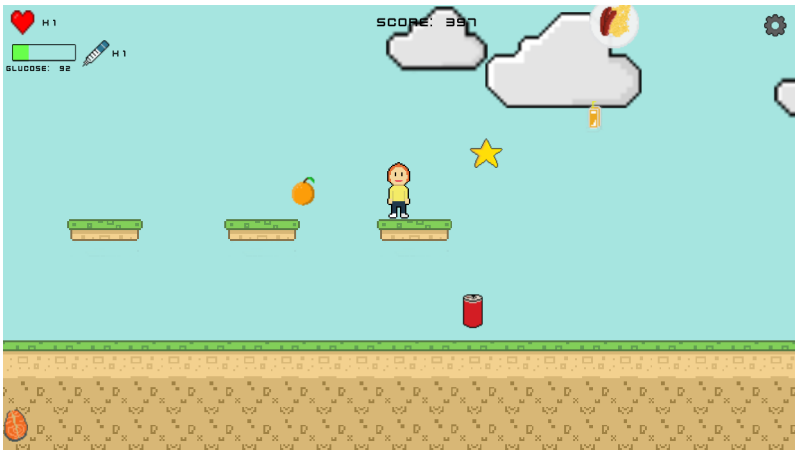


Figura 4.16: Imagen donde se observa el nivel del juego de plataformas

4.3.2. Desarrollo del juego de preguntas

Lo más prioritario en el juego de preguntas era tener una lista de niveles para poder acceder poder acceder a cada subnivel. Por ello, lo primero que implementamos fue una lista de niveles con los niveles que no estuvieran completos bloqueados (inicialmente todos menos el primero), y que se tuviera que acceder por orden a cada uno. Se muestra un candado sobre cada nivel que no esté desbloqueado, y si no completabas un nivel, no podías responder las preguntas de cada subnivel del siguiente, como se observa en la figura 4.17

Figura 4.17: Imagen que muestra el listado de niveles del juego de preguntas

Como inicialmente solo teníamos un nivel realizado, de primeras solo pudimos testear con un solo nivel. No obstante, después de implementar el bloqueo y desbloqueo de niveles, realizamos dos niveles más con las imágenes de comida que creamos. Cada nivel tiene unas preguntas fijas, pero éstas se generan con orden aleatorio.

Creamos también el menú de configuración que era igual que el del juego de plataformas. Se puede acceder a dicho menú desde el menú del juego de preguntas. Además, en los subniveles añadimos un botón para acceder al menú de configuración, con la particularidad de que el menú en los subniveles muestra una opción más: la de acceder a la lista de niveles, como se observa en la figura 4.14, la cual permite salir de cada subnivel.

Incorporamos, a su vez, una escena de "Nivel no superado", como se muestra en la figura 4.19 a la que se redirige al jugador en caso de que éste falle más de dos preguntas en el nivel. En el caso de que éste acierte más de tres preguntas, se mostrará otra escena similar a la de "Nivel no superado", llamada "Nivel superado", como se puede ver en la figura 4.20. En cada una de las escenas anteriormente mencionadas,

se permite al jugador volver al menú del juego de preguntas, volver a la lista de niveles, y repetir nivel o ir al siguiente nivel, respectivamente.

Debido a que la funcionalidad básica del juego de preguntas estaba implementada, lo siguiente que incorporamos fueron elementos más visuales. Se añadió una animación de un candado que aparece cuando se completa un nivel por primera vez. Además, se añadieron tres estrellas que iban asociadas a la puntuación obtenida en cada nivel, tanto en el listado de niveles (debajo del número del nivel), como en las escenas de "Nivel no superadoz "Nivel completado". Por cada nivel completado con cuatro aciertos, se muestran dos estrellas doradas y una gris. Si el nivel ha sido completado con cinco aciertos, se muestran tres estrellas doradas, como se observa en la figura 4.21. Si por el contrario, el nivel no ha sido superado porque el jugador ha obtenido más de dos fallos, se mostrará una estrella dorada y dos grises, como se puede ver en la figura 4.21. El jugador tiene la opción de repetir un nivel para mejorar una puntuación, y las estrellas se actualizarán en el listado de niveles.

Figura 4.18: Imagen que muestra el menú de configuración del juego de preguntas

Figura 4.19: Imagen donde se observa la pantalla de "Nivel no superado"del juego de preguntas

Figura 4.20: Imagen donde se puede ver la pantalla "Nivel superado"del juego de preguntas

Figura 4.21: Imagen donde se muestra un nivel del juego de preguntas superado

Figura 4.22: Imagen donde se puede ver la pantalla "Nivel superado"del juego de preguntas

Como detalle final, decidimos quitar el menú del juego de preguntas, ya que estaba entre el menú principal y el listado de niveles, y no añadía ninguna funcionalidad como tal. El resultado final fue que se accedía desde el menú principal al listado de preguntas. Además, en vez de tener dos escenas por si el jugador completaba o no completaba el nivel, reutilizamos una de las escenas, y cargamos textos diferentes en función de si el jugador superaba o no el nivel. Además añadimos la implementación de que cada tres niveles desbloqueados por el jugador, se le regala una vida para el juego de plataformas.

4.4. Mejora de la Interfaz de Usuario

Se realizó una mejora en la interfaz de usuario cuando cada juego en particular llevaba la mitad de su avance aproximadamente. Se hizo un lavado de cara a todos los menús y a las propias pantallas de los juegos. Usando imágenes de *Kenney Assets* (18), se cambió toda la estructura de cada menú, se eligieron colores que hacían buena combinación (19), y se cambió la fuente que se usaba hasta el momento por otra de Kenney.

Por una parte, el menú del juego de plataformas quedó tal y como se muestra en la figura 4.25. El resultado de cambiar la escena del "¡Perdiste!" se puede ver en la figura 4.26.

Varias personas de nuestro entorno probaron el juego de plataformas, y comprobaron que, sin información previa al respecto, la curva de dificultad inicial era muy grande. Por ello, añadimos dos escenas más en Unity que servían para aportar información sobre cómo jugar, cuánta glucosa bajaba la insulina, cuánta glucosa subía cada alimento, y la puntuación que se obtenía cuando conseguías una estrella. A la ventana de información se puede acceder pulsando la "i" que aparece en el menú del juego de plataformas. Ambas escenas se pueden ver en las figuras 4.23, 4.24.

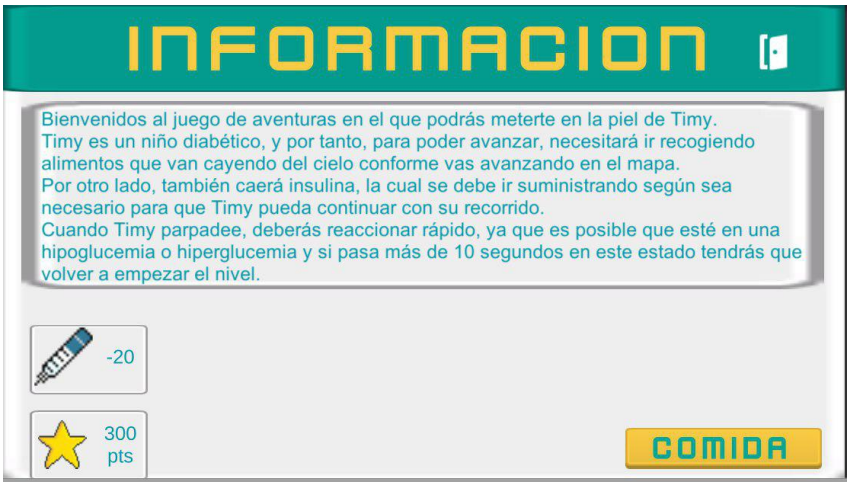


Figura 4.23: Imagen donde se muestra la escena de información general

Figura 4.24: Imagen donde podemos ver la escena de información sobre la comida

Figura 4.25: Imagen donde puede observar el menú final de plataformas

Figura 4.26: Imagen donde puede ver la pantalla del "Perdiste"del juego de plataformas

El menú de configuración también se cambió radicalmente. Se puede observar dicho cambio en la figura 4.27.

Figura 4.27: Imagen donde se puede observar el menú de configuración del juego de plataformas en mitad de una partida

Por otra parte, el juego de preguntas también mejoró su interfaz de usuario notablemente. Cambió tanto el listado de niveles, como la escena del “Nivel superado” o “Nivel no superado”. Este cambio se puede ver en las figuras 4.28, 4.21 y 4.22,

respectivamente.

Figura 4.28: Imagen donde vemos el listado de niveles del juego de preguntas

Además, después de incluir las imágenes que realizamos para el juego de preguntas, también mejoramos el aspecto de los subniveles, como se puede ver en la figura 4.29. El menú de configuración también se modificó, y quedó muy similar al del juego de plataformas, como se vio en 4.27, solo que con los colores que utilizamos para el juego de preguntas.

Para finalizar la mejora de interfaz de usuario, cambiamos el menú principal, que, combinando los colores de ambos juegos y poniéndolos algo más claros, quedaría como se puede observar en la figura 4.30.

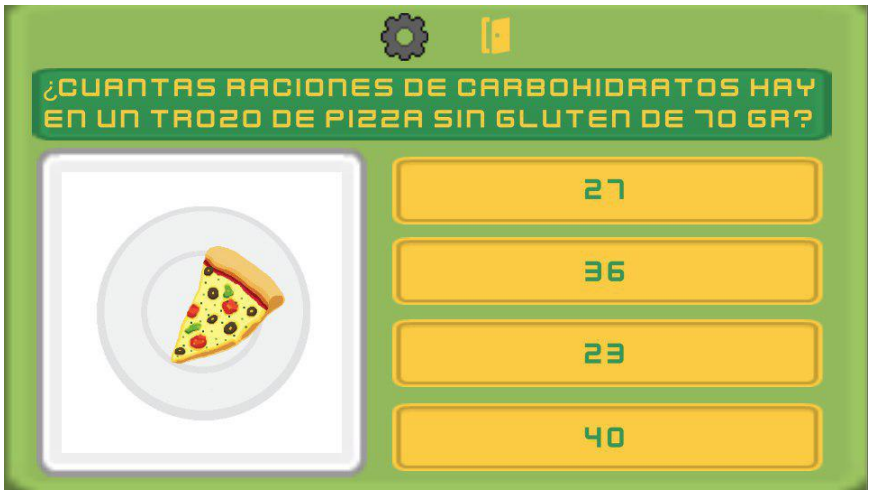


Figura 4.29: Imagen donde se puede observar un subnivel del juego de preguntas

Figura 4.30: Imagen donde vemos el menú principal después de la mejora de la interfaz de usuario

Capítulo 5

Resultados

The mind of the subject will desperately struggle to create memories where none exist...

Bioshock

En este capítulo exponemos los resultados finales obtenidos tras toda la etapa de desarrollo del videojuego. Además, mostramos una demo que describimos con detalle, que se encuentra en el siguiente enlace: `https://drive.google.com/drive/folders/1JZZgTuauvFnBIUOkNW3EStEzykh7YKqX?usp=sharing`

Al iniciar el juego, nos encontramos con el menú principal, donde ofrece la opción de entrar en un juego de aventuras, que es el de plataformas, en un juego de preguntas, o salir del propio juego. Si pulsas sobre el botón de "Aventura", la siguiente escena que carga es la del menú del juego de aventuras. Éste tiene un botón de información al lado del propio título del menú, en el que si pulsas, aparecerá una nueva ventana con información sobre dicho juego. En ella se explica el objetivo del juego y cómo jugar. Justo debajo de la explicación, aparece información de cuánta cantidad de glucosa baja la insulina, y la puntuación extra conseguida con la estrella. Además, si pulsas en el botón "Comida", te lleva a otra ventana de información sobre la comida y cuánta glucosa sube. Para volver atrás al menú, solo hace falta pulsar en la puerta que se muestra arriba a la derecha.

Por otra parte, estando en el menú de juego de plataformas, si pulsas en el icono de configuración situado arriba a la izquierda, aparece el menú de configuración, que tiene la barra para ajustar el volumen. Ésta es una funcionalidad que no hemos podido implementar, ya que el juego no tiene música. Volviendo al menú, podemos entrar en el nivel pulsando "Jugar". Como explicamos en el capítulo 4.3.1, cuando el jugador se queda sin vidas, no le queda otra que conseguirlas a través del juego de preguntas. Por ello, solo permitirá acceder al nivel en el caso de que el personaje tenga una o más vidas. De otro modo, aparecerá la escena de la lista de niveles del juego de plataformas.

También encontramos en dicho menú un botón "Ranking", en el que si pulsamos, se mostrará un *ranking* con las cinco mejores puntuaciones obtenidas por el jugador.

Si no hay suficientes puntuaciones, se mostrarán tres guiones en las posiciones vacías.

Una vez dentro del nivel del juego de plataformas, se irá generando comida e insulina que caen del cielo aleatoriamente. El jugador tendrá que ir avanzando por el nivel (corriendo y saltando), lo cual hará que la puntuación vaya subiendo, a la vez que la glucosa bajando. Por ello, tendrá que ir alimentándose con la comida que cae del cielo en función de los niveles de glucosa en sangre que el personaje tenga, e ir recolectando insulina por si en algún momento tuviera la glucosa demasiado elevada. Como bonus, podrá coger las estrellas que hay repartidas por el nivel, lo cual le dará 300 puntos extra.

Cada vez que el personaje pierde una vida, reaparece al principio del nivel y tanto la puntuación, como los niveles de glucosa de éste, se resetean (la glucosa empieza estando en 100). Cuando el personaje pierde todas las vidas, aparece la escena de "¡Perdiste!" del juego de plataformas. En esta, se muestra la mayor puntuación obtenida en la partida, y te da la opción de jugar al juego de preguntas para conseguir más vidas, volver al menú del juego de plataformas, o salir al menú principal.

Si vas directamente al juego de preguntas para conseguir vidas, se mostrará el listado de niveles y el avance del jugador hasta el momento. Se puede acceder al menú de configuración desde el botón que se muestra arriba a la izquierda. No obstante, para acceder a un nivel completo, solo hay que pulsar en uno de los niveles desbloqueados que aparecen. Si respondes menos de 4 preguntas correctamente, aparecerá una escena de "Nivel no superado", en la que se muestra la puntuación obtenida, unas estrellas que varían en función de dicha puntuación, y las opciones de ver niveles, repetir nivel, o salir al menú principal. Se puede repetir un nivel tantas veces como quieras, ya sea para superarlo, o para mejorar la puntuación obtenida previamente. Dicha puntuación se irá actualizando en la lista de niveles.

Cada vez que superes un nivel, aparecerá una escena de "Nivel superado", en la que, inicialmente, se verá una animación de un candado abriéndose. Una vez termine la animación, se podrá observar claramente el resultado obtenido, las estrellas asociadas a éste (dos o tres estrellas doradas, en función de los aciertos), y la opción de ver la lista de niveles, pasar al siguiente nivel, o salir al menú principal. Una vez completados tres niveles, se le sumará al personaje del juego de plataformas una vida.

Si sales del juego, se guardará tanto el progreso del juego de plataformas como el de preguntas en archivos externos. De momento, solo guarda las vidas del jugador, y los niveles completados del juego de preguntas, así como la puntuación obtenida en éstos.

Capítulo 6

Conclusiones y Posibles Mejoras

Happiness is only real when shared
Into the Wild

Este proyecto ha supuesto un reto tanto en el aprendizaje sobre la diabetes, como en la formación sobre C# y Unity. Dado que eran campos nuevos para nosotros, desde el principio nos fuimos encontrando con dificultades que tuvimos que sortear. Además, dado que la mayor parte de los sprites que usamos en este trabajo de fin de grado fueron realizados a mano.

No obstante, estamos satisfechos con el resultado obtenido tras el esfuerzo realizado durante todo un año. Creemos que este proyecto servirá para ayudar tanto a niños en los que ha debutado la diabetes como a adultos, y que facilitará la autogestión de su enfermedad.

A pesar de que los resultados de este proyecto fueron más que satisfactorios, hubo algunos requisitos que no nos dio tiempo a añadir, y algunos aspectos que se podrían mejorar. Por una parte, se podrían añadir más niveles al juego de preguntas y también al juego de plataformas. Además, añadir obstáculos extra en éste último, para así subir la dificultad del juego.

Por otra parte, añadir tanto música a los menús y la música principal del juego, como sonidos asociados a las acciones del personaje, (como saltar, correr, avisar cuando tiene los niveles de la glucosa en sangre elevados y cuando este muera), sería una retroalimentación que el jugador agradecería.

En cuanto a la estimación de raciones de carbohidratos, y cómo influye la insulina en el paciente de diabetes, se podría implementar el cálculo de la glucosa no lineal, y que el jugador pudiera meter parámetros de entrada como la sensibilidad a la insulina, para así darle más realismo al juego.

En definitiva, aunque se pueda continuar trabajando sobre este proyecto, el cual tiene potencial para causar un impacto positivo en pacientes con diabetes, y no solo en ellos, sino también en las personas que les rodean, que era al fin y al cabo nuestro objetivo principal.

Conclusions and Possible Improvements

Experience is simply the name we give our mistakes
Oscar Wilde

This project has been a big challenge because of the learning about diabetes, C# and Unity. Given that it was an unexplored area for us, we had to face some challenges during the development. Furthermore, almost all the sprites that we use at this project were made by us.

However, we are satisfied with the results achieved, which show the efforts made during this academic year. We believe that this project will help children as well as adults with diabetes, and that it will allow the self management of the desease.

In spite of the results were considered successful, there were some requirements that we didn't meet and some aspects that we could not improve since we could not find the time. On the one hand, more levels and gaming platforms could have been added to make more difficult the game. On the other hand, music and some sounds effects associated to the character would be a nice addition and a great feedback for the player. For instance when the character is jumping, running, hypoglycemic, or dying.

Regarding the carbohydrate ration estimation and how insulin affects the diabetes patient, the nonlinear glucose calculation could be implemented in the platform game. It would also be an extra implementation for this game that the player could enter the character's diabetes parameters, like glucose sensitivity or weight, for example, which could make the experience more realistic.

In sum, although this project still has pending implementations and some improvements to be made, it is likely to have a positive and beneficial impact on pacients with diabetes, as well as in other people around them, which was our main goal.

Contribuciones al proyecto

If you truly want to escape the ordinary, you'll simply need to keep evolving. Whether what you seek is above or below.

Durarara!

En este proyecto, salvo en la etapa del desarrollo de prototipos y diseño del software, hemos estado trabajando en su mayoría en las mismas implementaciones. Por ello, la mayor parte de nuestras aportaciones en este proyecto, sobre todo en las partes más cruciales de éste, han sido conjuntas. Cabe destacar también que cada uno ha revisado todo lo que el otro ha realizado en este proyecto, por lo cual no nos resulta tampoco fácil dividir el trabajo aportado en éste.

Por una parte, en el juego de preguntas, tenemos las siguientes contribuciones por parte de Mario:

- Creación de toda la base de datos en un .xml y el DAO que obtiene los datos de ésta.
- Implementación de la lógica del juego de preguntas: desde cargar las preguntas y respuestas guardadas en el DAO, hasta generar las respuestas aleatorias (además de la correcta) y pasar de una pregunta a otra.
- Consiguió que se pudiera pasar de un nivel a otro tanto en la lista de niveles como en la escena del "Nivel completado". También consiguió que se repitiera nivel en dicha escena en el caso de que no se hubiera completado, cuando aparece el "Nivel no completado".
- Revisión y corrección de errores de la implementación de las estrellas y los candados (ya sea en el listado de preguntas, como la animación que se muestra cuando completas un nivel).
- Creación y corrección de los errores de la escena del "Nivel completado"/"Nivel no completado".
- Contribuyó y corrigió errores en la implementación de que se pueda repetir un nivel ya completado.

- Calculó los gramos de carbohidratos que tenía cada comida del juego y lo adaptó en la base de datos.

A su vez, contribuyó en el juego de plataformas de la siguiente forma:

- Retocó los sprites de las plataformas que el personaje recorre en el juego de plataformas.
- Creó todos los *colliders* encargados de eliminar la comida de la pantalla, y de la muerte del personaje si éste se cae de las plataformas.
- Con la cámara de la escena, consiguió que el personaje no pudiera volver atrás en el nivel y que a su vez ésta persiguiese al personaje conforme se avanza por el escenario.
- Creación y revisión del algoritmo del cálculo lineal de la cantidad de glucosa en sangre del personaje en función de las comidas que éste ingiere, la insulina que se va suministrando y del ejercicio que hace.
- Implementación del botón que te permite consumir insulina (y funciona de un modo similar a consumir alimentos).
- Contribuyó y arregló errores en la implementación de que el personaje se ralentizara cuando tenía niveles elevados o bajos de glucosa en la sangre.
- Contribuyó y corrigió fallos en la implementación del parpadeo del personaje cuando éste tenía niveles excesivamente elevados o bajos de glucosa en la sangre.
- Creó y corrigió fallos de la escena del "Perdiste" cuando el personaje ya no tiene más vidas.
- Creación y modificación de las escenas de información en el juego de plataformas.
- Amplió el nivel, añadiendo más plataformas a éste.
- Creación y corrección de errores del ranking de las puntuaciones

Por último, sus contribuciones en las partes comunes a ambos juegos, y en la parte más general del proyecto fueron:

- A pesar de que cada uno hizo los prototipos por su cuenta para practicar con las herramientas que íbamos a utilizar durante el proyecto, los prototipos que se presentaron fueron los que Mario creó.
- Unificación de prototipos.
- Se encargó de toda la gestión del guardar y cargar partidas de ambos juegos.

- Modificación de los scripts que guardan la información de las partidas de cada juego, y del juego en general.
- Modificación de algunas funciones en el script encargado de cargar las escenas del juego.
- Contribuyó y revisó cada apartado de la memoria del proyecto.
- Revisó el código de cada juego.

Por otro lado, las contribuciones de Laura al juego de preguntas fueron las siguientes:

- Creación de estrellas y candados (incluyendo la animación del candado cuando completas un nivel) y su lógica correspondiente.
- Contribuyó y corrigió errores en la implementación de que se pueda repetir un nivel ya completado.
- Creación y corrección errores de la escena del "Nivel completado"/"Nivel no completado".
- Ayudó con el cálculo de los gramos de carbohidratos que tenía cada comida del juego.
- Corrección de errores en la implementación de que se pudiera pasar de un nivel a otro tanto en la lista de niveles como en la escena del "Nivel completado".
- Calculó los gramos de carbohidratos que tenía cada comida del juego y lo adaptó en la base de datos.

Además, contribuyó en el juego de plataformas de la siguiente forma:

- Creación y revisión del algoritmo del cálculo lineal de la cantidad de glucosa en sangre del personaje en función de las comidas que éste ingiere y del ejercicio que hace.
- Creación e implementación de la barra de glucosa del personaje.
- Creación e implementación del score del juego. Además, consiguió que se guardaran todos los score, y en la pantalla del "Perdiste" se muestre el mayor de todos ellos.
- Contribuyó y arregló errores en la implementación de que el personaje se ralentizara cuando tenía niveles elevados o bajos de glucosa en la sangre.
- Contribuyó y corrigió fallos en la implementación del parpadeo del personaje cuando éste tenía niveles excesivamente elevados o bajos de glucosa en la sangre.

- Creación del *spawner* de comida e insulina y de nubes en el nivel.
- Creó y corrigió fallos de la escena del "Perdiste" del juego.
- Creación y modificación de las escenas de información en el juego de plataformas.
- Creación y corrección de errores del ranking de las puntuaciones

Finalmente, sus contribuciones al proyecto a nivel general, y en la memoria, fueron:

- Creación de casi todos los sprites utilizados en los juegos.
- Unificación de prototipos.
- Creación y modificación de todos los session que tenemos.
- Creación y modificación del script encargado de cargar las escenas del videojuego.
- Creación y modificación de todos los menús del juego, incluido el de configuración.
- Modificación de la parte de la interfaz de usuario del videojuego.
- Contribuyó y revisó cada apartado de la memoria del proyecto.
- Revisó el código de cada juego.

Bibliografía

I still don't know what it really means to grow up.
However, if I happen to meet you, one day in the future,
by then, I want to become someone you can be proud to know.

5 Centimeters per Second - Makoto Shinkai

[1] M. Chan, "Informe mundial sobre la diabetes," *Organización Mundial de la Salud*, 2016.

[2] F. Díaz, "Los juegos serios y su potencial como dispositivos educativos," 2016, [Web; accedido el 15-08-2019]. [Online]. Available: http://www.eduforics.com/es/los-juegos-serios-y-su-potencial-como-dispositivos-educativos/

[3] A. C. Martín and C. T. Urquidi Aznar, "Juegos serios como instrumento facilitador del aprendizaje: evidencia empírica," *Opción*, vol. 31, no. 3, pp. 1201–1220, 2015.

[4] "Diabetes," 2019, [Web; accedido el 15-08-2019]. [Online]. Available: https://www.who.int/topics/diabetes_mellitus/es/

[5] "¿qué es la diabetes?" 2018, [Web; accedido el 9-08-2019]. [Online]. Available: https://www.clinicbarcelona.org/asistencia/enfermedades/diabetes/definicion

[6] "Qué es la diabetes," 2015, [Web; accedido el 10-08-2019]. [Online]. Available: https://www.fundaciondiabetes.org/prevencion/309/que-es-la-diabetes-2

[7] "Diabetes tipo 1," 2017, [Web; accedido el 10-08-2019]. [Online]. Available: https://www.niddk.nih.gov/health-information/informacion-de-la-salud/diabetes/informacion-general/que-es/diabetes-tipo-1

[8] VVAA, "El ejercicio y la diabetes tipo 1," 2013, [Web; accedido el 10-08-2019]. [Online]. Available: http://archives.diabetes.org/es/alimentos-y-actividad-fisica/condicion-fisica/el-ejercicio-y-la-diabetes-tipo-1.html

[9] "Qué es la diabetes," 2015, [Web; accedido el 10-08-2019]. [Online]. Available: https://www.fundaciondiabetes.org/infantil/176/que-es-la-diabetes-ninos

[10] "Diabetes tipo 2," 2019, [Web; accedido el 10-08-2019]. [Online]. Available: https://medlineplus.gov/spanish/diabetestype2.html

[11] "Guía diabetes tipo 1," 2016, [Web; accedido el 15-08-2019]. [Online]. Available: https://diabetes.sjdhospitalbarcelona.org/es/diabetes-tipo-1/debut/raciones-hidratos-carbono

[12] J. Hocking, *Unity in Action Multiplatform game development in C#*, 2nd ed. United States of America: Manning Publications, 2018.

[13] S. Rossel, *Object-Oriented Programming in C# Succinctly.* Syncfusion, 2016. [Online]. Available: https://www.syncfusion.com/ebooks/oop-csharp

[14] V. Sarcar, *Design Patterns in C#.* Apress, 2018. [Online]. Available: https://link-springer-com.bucm.idm.oclc.org/book/10.1007%2F978-1-4842-3640-6

[15] R. Nystrom, *Game Programming Patterns.* Genever Benning, 2014, [Web; accedido por última vez el 25-08-2019]. [Online]. Available: http://gameprogrammingpatterns.com/

[16] J. I. Hidalgo, J. M. Colmenar, G. Kronberger, S. M. Winkler, O. Garnica, and J. Lanchares, "Data based prediction of blood glucose concentrations using evolutionary methods," *Journal of medical systems*, vol. 41, no. 9, p. 142, 2017.

[17] "Raciones de hidratos de carbono," [Web; accedido el 20-08-2019]. [Online]. Available: https://diabetes.sjdhospitalbarcelona.org/es/diabetes-tipo-1/raciones-hidratos-carbono

[18] Kenney, "Ui pack," [Web; accedido el 17-07-2019]. [Online]. Available: https://kenney.nl/assets/ui-pack

[19] "Adobe color," [Web; accedido el 17-07-2019]. [Online]. Available: https://color.adobe.com/es/explore

Until they become conscious they will never rebel,
and until after they have rebelled they cannot become conscious.

1984
George Orwell

These people don't see that if you encourage totalitarian methods,
the time may come when they will be used against you instead of for you.

Animal Farm
George Orwell